Diffusion-free Leafnet algorithm for multidimensional nonlinear hyperbolic systems and numerical solution with neural networks

Emmanuel LORIN[a,b], Arian NOVRUZI[c]

[a] *School of Mathematics and Statistics, Carleton University, Ottawa, Canada, K1S 5B6*
[b] *Centre de Recherches Mathématiques, Université de Montréal, Montréal, Canada, H3T 1J4*
[c] *Department of Mathematics and Statistics, University of Ottawa, Ottawa, Canada*

Abstract

This paper is devoted to the non-diffusive computation of solutions to N-dimensional hyperbolic systems of conservation laws (HCL) and more specifically the accurate approximation of simple waves. We develop the *leafnet* algorithm, which relies on: i) a reformulation of HCL as a coupled system involving smooth solutions, and ii) an accurate physics informed algorithms approximating smooth functions. An error estimate analysis and two-dimensional numerical experiments are proposed to illustrate the presented strategy.

Keywords: Hyperbolic equations; conservation laws; weak solutions; optimization; neural network; machine learning

1. Introduction

This paper is devoted to the neural network-based solution of multidimensional / N-dimensional nonlinear hyperbolic conservation laws (HCL), see [1, 2, 3]. The use of neural networks for solving (parameterized) PDE has recently become very popular, as beyond the use of automatic differentiation, they allow for the inclusion of external (experimental or numerical) data in the solver. Among the existing techniques the most popular are the Physics Informed Neural Network (PINN) methods [4, 5, 6], which can be seen as a generalization of the pioneering work in [7]. Let us also cite alternative methods presented in [8, 9, 10].

In this paper, we are particularly interested in the approximation of simple waves, more precisely shock and rarefaction waves, which is known to be a difficult numerical problem with standard computational methods, such as finite volume methods [11, 12, 13, 14]. The latter rely on approximate Riemann solvers, and usually produce some artificial diffusion. Increasing the order of approximation allows for a reduction of the artificial diffusion, but also leads to stability issues, requiring the use of slope limiters [15]. On this other hand, in

Email addresses: elorin@math.carleton.ca (Emmanuel LORIN), novruzi@uottawa.ca (Arian NOVRUZI)

order to approximate weak non-smooth solutions alternative methods based on variational formulation of PDEs, such as Deep Ritz [16] or Variational-PINN [17], have been developed.

PINN-like methods allow for the approximation of smooth solutions to HCL (including rarefaction waves), avoiding to a certain extent stability issues usually due to time-stepping. Indeed, within PINN algorithms, IVP problems are often reformulated and approximated as a BVP, where the initial condition is treated as a boundary condition. Stability issues may however be moved to the optimization algorithm. In order to avoid a null solution, the integration time has to be taken not too large, which represents a drawback of these methods.

As for the shock waves, their approximation with classical neural network algorithms, such as PINN in [4, 5, 6], is known to be very inaccurate. This is mainly due to the fact that shock waves are non-smooth solutions, defined by the weak formulation of the hyperbolic conservation laws, and that similarly to finite difference methods, neural network algorithms, such as PINN methods, approximate PDEs in their strong form. As a consequence, a direct use of (smooth) neural networks only generates smooth solutions (hence corresponding to regularized shocks).

In this paper, we develop an algorithm allowing to accurately approximate shock waves using neural networks by: i) approximating smooth solutions, ii) approximating the Rankine-Hugoniot jump condition, and iii) reconstructing the overall piecewise smooth solution. The proposed method can be seen as a multidimensional extension of the algorithm which was developed in [18] for one-dimensional systems, and which largely relies on the structure of solutions as well as mathematical properties of HCL. We will focus on the approximation of shock waves for N-dimensional hyperbolic conservation laws, including their generation from smooth Cauchy data. We also will design self-similar neural networks allowing for an accurate approximation of rarefaction waves. Finally we will extend our algorithm to the systems of HCL. The extension of the proposed strategy to more complex problems, such as wave-interactions, overturning or closed surfaces, will covered in a forthcoming paper.

The paper is organized as follows. In Section 2, we derive and analyze our method for N-dimensional scalar conservation laws. Namely, in Subsection 2.1, we discuss the general strategy for N-dimensional scalar hyperbolic equations using the mathematical structure of the weak solutions. More precisely, in Subsection 2.1.1 we present the associated algorithm. In Subsection 2.1.2 we propose a mathematical justification of our algorithm, followed by a detailed mathematical analysis of the error for shock wave solutions in Subsection 2.1.3. In Subsection 2.2 we discuss the formation of shock waves and in Subsection 2.3 an algorithm for generating rarefaction waves is derived. The extension our method to N-dimensional systems of HCL is presented in Section 3. Section 4 is devoted to numerical experiments, and concluding remarks are proposed in Section 5.

We finish this introduction by recalling some basic concepts on neural networks in N-dimensions. Let $l \in \mathbb{N}$, $n_i \in \mathbb{N}$, $i = 0, 1, \ldots, l$, with $n_0 = N$, and $\sigma : \mathbb{R} \mapsto \mathbb{R}$ a given nonlinear function, called activation function. For $\theta \in \prod_{i=1}^{l} \mathbb{R}^{n_i \times (n_{i-1}+1)}$, we write

$$\begin{aligned} \theta &= (\theta^i), \quad \theta^i = (w^i, b^i), \quad i = 1, \ldots, l, \quad \textit{where} \\ & \quad w^i(\cdot, \cdot) \in \mathbb{R}^{n_i \times n_{i-1}}, \end{aligned}$$

$$b^i(\cdot) \in \mathbb{R}^{n_i}.$$

A (fully connected) network in $\mathbb{R}^N$ with architecture $A = [n_0, n_1, \dots, n_l]$ is a function of the form

$$\mathbf{u} : (\theta; x) \in \prod_{i=1}^{l} \mathbb{R}^{n_i \times (n_{i-1}+1)} \times \mathbb{R}^N \mapsto \mathbf{u}(\theta; x) \in \mathbb{R}^{n_l}, \tag{1a}$$

$$\mathbf{u}(\theta; x) = w^l \cdot \sigma(w^{l-1} \cdot (\cdots \sigma(w^2 \cdot \sigma(w^1 \cdot x + b^1) + b^2) \cdots) + b^{l-1}) + b^l, \tag{1b}$$

where σ applies component-wise. Given an architecture A, the set of all parameters θ associated to A is denoted by Θ_A. The set of all network functions associated to parameters in Θ_A is denoted by $\mathbf{N}_A$. All along this paper, the network functions will be denoted by boldface letters.

2. Scalar conservation laws in N dimensions

In this section we consider a $N \geqslant 1$ dimensional nonlinear scalar hyperbolic conservation law in $Q := \Omega \times [0, T]$ for some (un)bounded open smooth set $\Omega \subseteq \mathbb{R}^N$. Denoting $\nabla = (\partial_1, \dots, \partial_N) = (\partial_{x_1}, \dots, \partial_{x_N})$, the problem we consider is

$$\partial_t u + f(u) \cdot \nabla u = 0, \qquad \text{in } Q, \tag{2a}$$

$$u(\cdot, 0) = u_0(\cdot), \qquad \text{on } \Omega, \tag{2b}$$

where $u : (x, t) := (x_1, \dots, x_N, t) \in Q \to u(x, t) \in \mathbb{R}$, $f(u) = (f_1(u), \dots, f_N(u)) =: F'(u)$, $F \in C^2(\mathbb{R}; \mathbb{R}^N)$.

We impose incoming flux boundary conditions - that is boundary conditions are imposed where characteristic lines are incoming. For simplicity of presentation, we will not discuss these boundary conditions and we refer to [19] for details. For simplicity of presentation, here we will consider $\Omega = \prod_{i=1}^N (a_i, b_i)$, or $\Omega = \mathbb{R}^N$. The latter case keeps the full generality of the problem, if we extend smoothly u_0 to a function with compact support, and reducing the time T.

2.1. Shock waves and the principle of the leafnet method

Let $\Omega = \prod_{i=1}^N (a_i, b_i)$. Given two functions $u_0^{\pm} \in C^1(\Omega)$, and a C^1-surface $\Gamma_0 \subset \Omega$, parameterized by $g_0 \in C^1(\Omega')$, $\Omega' := (a_1, b_1) \times \cdots \times (a_{N-1}, b_{N-1})$,

$$\Gamma_0 = \{(x', x_N) \in \Omega,\ x_N = g_0(x'), \quad x' = (x_1, \dots, x_{N-1}) \in \Omega'\},$$

where the direction x_N is chosen arbitrarily, without loss of generality. We define $\Omega^{\pm} \subset \Omega$ by

$$\Omega^+ = \{(x', x_N) \in \Omega,\ x_N > g_0(x')\}, \quad \Omega^- = \{(x', x_N) \in \Omega,\ x_N < g_0(x')\}, \tag{3}$$

and the $C^1(\Omega)$-piecewise function u_0 by

$$u_0(x) = u_0(x', x_N) = \begin{cases} u_0^+(x), & x_N > g_0(x'), \\ u_0^-(x), & x_N < g_0(x'). \end{cases} \tag{4}$$

The shock of (2) with u_0 given by (4) is identified as the surface Γ in Q, where the Rankine-Hugoniot condition is satisfied. We assume that the evolution of the shock wave of (2) is parameterized by $x_N = g(x', t)$, i.e.

$$\Gamma \;=\; \big\{(x', g(x', t), t) \in Q \;:\; (x', t) \in Q' := \Omega' \times (0, T)\big\}. \tag{5}$$

Our objective is in particular to identify the function g. Let $\nu = (\nu_1, \ldots, \nu_N, \nu_t)$ be the unit normal vector on Γ oriented on the direction of x_N axis given by

$$\nu = \frac{\nabla_{(x,t)}(x_N - g(x', t)}{|\nabla_{(x,t)}(x_N - g(x', t))|} = \frac{(-\partial_1 g, \ldots, -\partial_{N-1} g, 1, -\partial_t g)}{\sqrt{1 + |\partial_1 g|^2 + \cdots + |\partial_{N-1} g|^2 + |\partial_t g|^2}}, \tag{6}$$

The Rankine-Hugoniot jump condition on Γ in terms of $u^\pm$ reads

$$\nu_t [\![u]\!] + [\![F(u)]\!] \cdot (\nu_1, \ldots, \nu_N) \;=\; 0, \tag{7}$$

where in general, for any function v, the jump operator $[\![\cdot]\!]$ on Γ is defined by

$$[\![v]\!] = \lim_{h \to 0} (v(x', g(x') + h^2) - v(x', g(x') - h^2)). \tag{8}$$

Then by rearranging (7), we deduce that g satisfies

$$\begin{aligned} \partial_t g + \frac{[\![F_1(u)]\!]}{[\![u]\!]} \partial_1 g + \cdots + \frac{[\![F_{N-1}(u)]\!]}{[\![u]\!]} \partial_{N-1} g =: & \\ \partial_t g + a_1(g, x', t) \partial_1 g + \cdots + a_{N-1}(g, x', t) \partial_{N-1} g = b(g, x', t) & \qquad (9a) \\ := \frac{[\![F_N(u)]\!]}{[\![u]\!]}, & \\ g(\cdot, 0) = g_0(\cdot)\,. & \qquad (9b) \end{aligned}$$

Hereafter we rewrite the jumps $[\![\cdot]\!]$ in (9) in a form that will be used for the analysis and the error estimate in the following sections. First, we consider the following problems (with appropriate incoming flux boundary conditions, or no boundary conditions if $u_0^\pm$ are extended smoothly to functions with compact support),

$$\partial_t u^\pm + f(u^\pm) \cdot \nabla u^\pm = 0, \quad \text{in } Q, \tag{10a}$$

$$u^\pm(\cdot, 0) = u_0^\pm, \quad \text{on } \Omega. \tag{10b}$$

Subject to taking T smaller, we assume that there is no shock formation for both IBVPs (10) for $t \leqslant T$. Following [20, Theorem 3.1], this holds if both $\nabla \cdot f(u_0^\pm)$ take negative values and $\nabla \cdot f(u_0^\pm)$ are bounded. In such a case, T can be defined as

$$T = -\frac{1}{\min\{\nabla \cdot f(u_0^\pm(\alpha)),\ \alpha \in \mathbb{R}^N\}}, \quad \text{where } \nabla \cdot f(u_0^\pm) = \sum_{i=1}^{N} f_i'(u_0^\pm) \partial_i u_0^\pm, \tag{11}$$

From the uniqueness of classical solutions to (2) we have $u = u^+$ above Γ and $u = u^-$ below Γ (in the x_N direction). Then for $i = 1, \dots, N$ we have

$$\begin{aligned}
[\![F_i(u)]\!] &= F_i(u^+(x', g(x', t), t)) - F_i(u^-(x', g(x', t), t)) \\
&= (u^+(x', g(x', t), t) - u^-(x', g(x', t), t)) \\
&\quad \cdot \int_0^1 f_i(u^-(x', g(x', t), t) + s(u^+(x', g(x', t), t) - u^-(x', g(x', t), t)))ds \\
&= [\![u]\!] \int_0^1 f_i(u^- + s[\![u]\!])ds \\
&=: [\![u]\!] \begin{cases} a_i(x', t), & i = 1, \dots, N-1, \\ b(x', t), & i = N. \end{cases}
\end{aligned}$$

This allows to rewrite the coefficients a_i and b in (9) as follows

$$a = (a_1, \dots, a_{N-1}),$$

$$a_i(g, x', t) = \int_0^1 f_i(u^-(x', g, t) + s(u^+(x', g, t) - u^-(x', g, t)))ds, \tag{12a}$$

$$b(g, x', t) = \int_0^1 f_N(u^-(x', g, t) + s(u^+(x', g, t) - u^-(x', g, t)))ds. \tag{12b}$$

We will use (9) with the form (12) of coefficients a, b for estimating the error of the method. It will be helpful in the following to write (9) in operator form as follows

$$K[g] = (K_i[g], K_b[g]) := (\partial_t g + a \cdot \nabla_{x'} g - b, g(\cdot, 0) - g_0(\cdot)) = (0, 0), \tag{13}$$

where a and b are given by (12).

Hence, "solving" the shock Γ of (2) is equivalent to solve the $N-1$ dimensional IVP (9). The method, that we will refer as *leafnet method*, consists in:

1) *Solving (independently) the two IBVP* (10)*, which provide* u^- *and* u^+ *in* Q.
2) *Solving the Rankine-Hugoniot* (9)*, which provides* g *in* Q'.
3) *Reconstructing the solution from the initial IBVP* (10) *for* $(x', x_N, t) \in Q$*, as follows*

$$u(x', x_N, t) = \begin{cases} u^-(x', x_N, t), & \text{if } x_N < g(x', t)\,, \\ u^+(x', x_N, t), & \text{if } x_N > g(x', t)\,. \end{cases} \tag{14}$$

Such a construction of the shock waves is presented in [20] for $N = 1, 2$. Here we extend it in dimension N. Furthermore, we will provide precise C^1 estimates for the solutions $u^\pm$, which will be useful for the error estimate of the network approximation of the shock wave.

2.1.1. Neural network algorithm: the leafnet algorithm

The leaflnet method, presented in Subsection 2.1, see (14), benefits from the accurate approximation of classical solutions by PINN algorithms, hence avoiding spurious regularization of discontinuous waves. We implement it by using a combination of PINN algorithms. This algorithm will be referred as the *leafnet* algorithm.

We introduce two neural networks $\mathbf{u}^{\pm}(x,t)$, $x=(x_1,\dots,x_N)$ for solving (10). In addition, we introduce one neural network $\mathbf{g}(x',t)$ for solving (9). More precisely we proceed as follows.

1. *Approximation of* (10). We introduce the operator

$$L^{\pm}[u]=(L_i^{\pm}[u],L_b^{\pm}[u]):=(\partial_t u+f(u)\cdot\nabla u,u(\cdot,0)-u_0^{\pm}). \tag{15}$$

Given an architecture A in Q, with parameters Θ_A, for $k,m\in\mathbb{N}_0$ and $\mathbf{u}\in\mathbf{N}_A$ define

$$\boldsymbol{\mathcal{L}}^{\pm}[\mathbf{u}]=\sum_{\substack{\alpha\in\mathbb{N}_0^N,\beta\in\mathbb{N}_0\\ |\alpha|+\beta\leqslant k}}\int_0^T\int_\Omega|D_x^\alpha D_t^\beta L_i^{\pm}[\mathbf{u}]|^2dxdt+\sum_{\alpha\in\mathbb{N}_0^N,|\alpha|\leqslant m}\int_\Omega|D_x^\alpha L_b^{\pm}[\mathbf{u}]|^2dx. \tag{16a}$$

We refer to [21, 22] for discussions on the use of Sobolev norms to define loss functions. Next we look for $\mathbf{u}^{\pm}\in\mathbf{N}_A$ the solution to

$$\mathbf{u}^{\pm}=\operatorname{argmin}\{\boldsymbol{\mathcal{L}}^{\pm}[\mathbf{u}],\ \ \mathbf{u}\in\mathbf{N}_A\}. \tag{17}$$

Note that in (17), the optimization is made with respect to the weights $\theta\in\Theta_A$, and $\mathbf{u}^{\pm}$ represents the network which corresponds to the optimal parameters $\theta^{\pm}\in\Theta_A$. We will show that under specific conditions, $\mathbf{u}^{\pm}$ approximates $u^{\pm}$.

2. *Solution to the IVP* (9). We consider an architecture A' in Ω', and denote by $\Theta_{A'}$, resp. $\mathbf{N}_{A'}$, its associated set of parameters and network functions. For $\mathbf{h}\in\mathbf{N}_{A'}$ we define

$$\mathbf{K}[\mathbf{h}]=(\mathbf{K}_i[\mathbf{h}],\mathbf{K}_b[\mathbf{h}])=(\partial_t\mathbf{h}+\mathbf{a}\cdot\nabla_{x'}\mathbf{h}-\mathbf{b},\mathbf{h}(\cdot,0)-g_0(\cdot)), \tag{18}$$

where

$$\mathbf{a}=(\mathbf{a}_1,\dots,\mathbf{a}_{N-1}),$$

$$\mathbf{a}_i(\mathbf{h},x',t)=\int_0^1 f_i(\mathbf{u}^-(x',\mathbf{h},t)+s(\mathbf{u}^+(x',\mathbf{h},t)-\mathbf{u}^-(x',\mathbf{h},t)))ds, \tag{19a}$$

$$\mathbf{b}(\mathbf{h},x',t)=\int_0^1 f_N(\mathbf{u}^-(x',\mathbf{h},t)+s(\mathbf{u}^+(x',\mathbf{h},t)-\mathbf{u}^-(x',\mathbf{h},t)))ds, \tag{19b}$$

and $u^{\pm}$ are defined in (10). For $p,q\in\mathbb{N}_0$ and $\mathbf{h}\in\mathbf{N}_{A'}$ define

$$\boldsymbol{\mathcal{K}}[\mathbf{h}]=\sum_{\substack{\alpha\in\mathbb{N}_0^N,\beta\in\mathbb{N}_0\\ |\alpha|+\beta\leqslant p}}\int_0^T\int_{\Omega'}|D_x^\alpha D_t^\beta\mathbf{K}_i[\mathbf{h}]|^2dxdt+\sum_{\alpha\in\mathbb{N}_0^N,|\alpha|\leqslant q}\int_{\Omega'}|D_x^\alpha\mathbf{K}_b[\mathbf{h}]|^2dx\,, \tag{20a}$$

and define $\mathbf{g}\in\mathbf{N}_{A'}$ by

$$\mathbf{g}=\operatorname{argmin}\{\boldsymbol{\mathcal{K}}[\mathbf{h}],\ \ \mathbf{h}\in\mathbf{N}_{A'}\}, \tag{21}$$

3. Reconstruction of the approximate solution $\mathbf{u}$ to (2), for $(x', x_N, t) \in Q$

$$\mathbf{u}(x', x_N, t) = \begin{cases} \mathbf{u}^-(x', x_N, t), & \text{if } x_N < \mathbf{g}(x', t)\,, \\ \mathbf{u}^+(x', x_N, t), & \text{if } x_N > \mathbf{g}(x', t)\,. \end{cases} \tag{22}$$

In this paper, we have not imposed any hard constraints forcing the solver to only approximate entropic waves only. Entropy conditions, such as (23), could easily be included within the loss but would deteriote the training of the neural networks. Let us add, that entropy questions was shortly addressed in [18].

2.1.2. Theoretical justification of Algorithm 2.1.1

In this section we will justify rigorously the leaf method (14). The existence part of the following result has been presented in [20] in dimension two. We extend it in dimension N and provide some L^∞-estimates which will be important for estimating the error on approximate shock waves.

To avoid the technicalities related to the boundary conditions, all along this section we assume that $u^\pm$ are with compact support and we take $\Omega = \mathbb{R}^N$, $\Omega' = \mathbb{R}^{N-1}$.

Theorem 2.1. *Let $F = (F_1, \dots, F_N) \in C^2(\mathbb{R}, \mathbb{R}^N)$, $f = (f_1, \dots, f_N) := F'$, with f_N increasing, $g_0 \in C^1_b(\mathbb{R}^{N-1})$ and $\Gamma_0 = \{(x', g_0(x')),\ x' \in \mathbb{R}^{N-1}\}$ the graph of g_0 with normal vector $\nu^0 = (\nu^0_1, \dots, \nu^0_N)$ oriented toward x_N axis. Let also $u_0^+, u_0^- \in C^1(\mathbb{R}^N)$ with compact support satisfying the following entropy condition (see [20])*

$$f_N(u_0^-(x)) > \frac{[\![F_N(u_0(x))]\!]}{[\![u_0(x)]\!]} > f_N(u_0^+(x)), \quad \forall x \in \Gamma_0, \tag{23}$$

where $[\![\varphi(u)]\!] := \varphi(u_0^+) - \varphi(u_0^-)$ on Γ_0 for any function φ, see (8), and define u_0 as in (4). If $T > 0$ is small enough then the following statements hold:
(i) Problems (10) have unique solutions $u^+, u^- \in C^1(\mathbb{R}^N \times [0,T])$. Furthermore,

$$\|u^\pm\|_{L^\infty(\mathbb{R}^N \times [0,T])} \leqslant \|u_0^\pm\|_{L^\infty(\mathbb{R}^N)} := \rho, \tag{24}$$

$$\begin{aligned} \|\partial_t u^\pm\|_{L^\infty(\mathbb{R}^N \times [0,T])} + \|\nabla u^\pm\|_{L^\infty(\mathbb{R}^N \times [0,T])} &\leqslant \|\nabla u_0\|_{L^\infty(\mathbb{R}^N)} \\ &+ C(\|\partial_z f\|_{L^\infty(B_\rho \times \mathbb{R}^N \times \mathbb{R})}, \|\nabla_x f\|_{L^\infty(B_\rho \times \mathbb{R}^N \times \mathbb{R})}). \end{aligned} \tag{25}$$

(ii) The problem (9) has a unique piecewise $C^1(\mathbb{R}^{N-1} \times [0,T])$ weak solution, and (2) has a unique piecewise $C^1(\mathbb{R}^N \times [0,T])$ solution u, with a C^1-discontinuity surface Γ with equation $x_N = g(x', t)$, which can be constructed as in 3), (14).

The proof of this theorem is based on the following general result for the quasi-linear first order PDE problems:

$$\partial_t u + \sum_{i=1}^N f_i(u, x, t)\partial_i u = r(u, x, t), \qquad \text{in } \mathbb{R}^N \times (0, T), \tag{26a}$$

$$u(\cdot, 0) = u_0(\cdot), \quad \text{on } \mathbb{R}^N\,, \tag{26b}$$

where $u : (x,t) \in \mathbb{R}^N \times [0,T] \mapsto u(x,t) \in \mathbb{R}$, and f, r satisfy:

$$f = (f_1, \dots, f_N), \quad f_i = f_i(z,x,t) \in C^1(\mathbb{R} \times \mathbb{R}^N \times \mathbb{R}), \tag{27a}$$

$$\forall \rho > 0,\ \exists C(\rho) \geqslant 0 : \quad \|f\|_{L^\infty(B_\rho \times \mathbb{R}^N \times \mathbb{R})} + \|\nabla_{(z,x)} f\|_{L^\infty(B_\rho \times \mathbb{R}^N \times \mathbb{R})} \leqslant C(\rho) < \infty, \tag{27b}$$

and

$$r = r(z,x,t) \in C^1(\mathbb{R} \times \mathbb{R}^N \times \mathbb{R}), \tag{28a}$$

$$\exists C_r \geqslant 0 : \quad \|r(z,\cdot,\cdot) - r(0,\cdot,\cdot)\|_{L^\infty(\mathbb{R}^N \times \mathbb{R})} \leqslant C_r |z|, \tag{28b}$$

$$\forall \rho > 0,\ \exists C(\rho) \geqslant 0 : \quad \|r\|_{L^\infty(B_\rho \times \mathbb{R}^N \times \mathbb{R})} + \|\nabla_{(z,x)} r\|_{L^\infty(B_\rho \times \mathbb{R}^N \times \mathbb{R})} \leqslant C(\rho) < \infty, \tag{28c}$$

where B_ρ is the ball in $\mathbb{R}$ with radius ρ and center at the origin and $C(\rho)$ is a constant depending on ρ (and f and r, but having f and r fixed we omit the dependence on f and r).

The following result related to the problem (26) is very important for proving Theorem 2.1. The result is classical, but as we did not find it firmly stated in the literature, for sake of completeness and especially for establishing the estimates that we will need, we present it in detail.

Theorem 2.2. *Let $u_0 \in C^1(\mathbb{R}^N)$ with compact support, and f, resp. r, as in* (27)*, resp* (28)*. Then there exists $T > 0$ such that* (26) *has a unique classical solution $u \in C^1(\mathbb{R}^N \times [0,T])$ satisfying*

$$\|u\|_{L^\infty(\mathbb{R}^N \times [0,T])} \leqslant (\|u_0\|_{L^\infty(\mathbb{R}^N)} + T\|r(0,\cdot)\|_{L^\infty(\mathbb{R}^N \times \mathbb{R})}) e^{TC_r} =: \rho, \tag{29}$$

$$\begin{aligned} \|\partial_t u\|_{L^\infty(\mathbb{R}^N \times [0,T])} + \|\nabla u\|_{L^\infty(\mathbb{R}^N \times [0,T])} &\leqslant \|\nabla u_0\|_{L^\infty(\mathbb{R}^N)} \\ &+ C(\|\nabla_{(z,x)} f\|_{L^\infty(B_\rho \times \mathbb{R}^N \times \mathbb{R})}, \|\nabla_{(z,x)} r\|_{L^\infty(B_\rho \times \mathbb{R}^N \times \mathbb{R})}), \end{aligned} \tag{30}$$

where the constant $C(\cdot,\cdot)$ can be estimated by using (35)*,* (37)*.*

Proof. The characteristic equations of (26) are given by

$$X(t;\alpha) = \alpha + \int_0^t f(z(s;\alpha), X(s,\alpha), s) ds, \quad t \geqslant 0, \qquad X(0,\alpha) = \alpha,\ \alpha \in \mathbb{R}^N, \tag{31a}$$

$$z(t;\alpha) = u_0(\alpha) + \int_0^t r(z(s;\alpha), X(s,\alpha), s) ds, \quad t \geqslant 0, \qquad z(0,\alpha) = u_0(\alpha). \tag{31b}$$

where $X(t;\alpha) = (X_1(t;\alpha), \dots, X_N(t;\alpha))$. By a standard fixed point argument, it is easy to show that the equations (31) have a global C^1 unique solution. As f and r are C^1, form classical results we know that $X(t;\alpha)$ and $z(t;\alpha)$ are C^1 in $(t,\alpha) \in \mathbb{R} \times \mathbb{R}^N$ and C^2 in $t \in \mathbb{R}$.

Now we construct the solution $u(x,t)$ by using $z(t;\alpha)$ and $X(t;\alpha)$. For this we need to show that the map $(\alpha,t) \in \mathbb{R}^N \times [0,T] \mapsto (X(t;\alpha),t) \in \mathbb{R}^N \times [0,T]$ is C^1-invertible, where $T > 0$ is small enough. This requires estimates for X, z and their derivatives near $(\alpha,0) \in \mathbb{R}^{N+1}$.

From (31b) and assumptions (28b), (28c) we get

$$|z(t,\alpha)| \;\leqslant\; \|u_0\|_{L^\infty(\mathbb{R}^N)} + t\|r(0,\cdot)\|_{L^\infty(\mathbb{R}^N\times\mathbb{R})} + C_r\int_0^t |z(s,\alpha)|ds,$$

which combined with Grönwall's inequality gives

$$\|z\|_{L^\infty([0,T]\times\mathbb{R}^N)} \;\leqslant\; (\|u_0\|_{L^\infty(\mathbb{R}^N)} + T\|r(0,\cdot)\|_{L^\infty(\mathbb{R}^N\times\mathbb{R})})e^{TC_r} =: \rho. \tag{32}$$

This estimate and (31a) imply

$$\|X - I_N\|_{L^\infty([0,T]\times\mathbb{R}^N)} \leqslant T\|f\|_{L^\infty(B_\rho\times\mathbb{R}^N\times\mathbb{R})}. \tag{33}$$

The equations (31) in combination with (32) yield

$$\|\partial_t X\|_{L^\infty([0,T]\times\mathbb{R}^N)} \leqslant \|f\|_{L^\infty(B_\rho\times\mathbb{R}^N\times\mathbb{R})}, \tag{34a}$$
$$\|\partial_t z\|_{L^\infty([0,T]\times\mathbb{R}^N)} \leqslant \|r\|_{L^\infty(B_\rho\times\mathbb{R}^N\times\mathbb{R})}. \tag{34b}$$

By differentiating with respect to α the equations (31) gives

$$\begin{aligned}\begin{bmatrix} \partial_{\alpha_i}X(t;\alpha) \\ \partial_{\alpha_i}z(t;\alpha)\end{bmatrix} &= \begin{bmatrix} e^i \\ \partial_{\alpha_i}u_0(\alpha)\end{bmatrix} \\ &+ \int_0^t \begin{bmatrix} \partial_x f(z(s;\alpha),X(s;\alpha),s), & \partial_z f(z(s;\alpha),X(s;\alpha),s) \\ \partial_x r(z(s;\alpha),X(s;\alpha),s), & \partial_z r(z(s;\alpha),X(s;\alpha),s)\end{bmatrix}\cdot\begin{bmatrix} \partial_{\alpha_i}X(t;\alpha) \\ \partial_{\alpha_i}z(t;\alpha)\end{bmatrix} ds,\end{aligned}$$

with e^i the ith vector of the canonical base in $\mathbb{R}^N$. It implies that

$$\begin{aligned}&|\partial_{\alpha_i}X - e^i| + |\partial_{\alpha_i}z - \partial_{\alpha_i}u_0| \\ \leqslant\;& t\left(\sum_{h=f,r}\|\partial_x h\|_{L^\infty(B_\rho\times\mathbb{R}^N\times\mathbb{R})} + \sum_{h=f,r}\|\partial_z h\|_{L^\infty(B_\rho\times\mathbb{R}^N\times\mathbb{R})}\|\nabla u_0\|_{L^\infty(\mathbb{R}^N)}\right) \\ +\;& \sum_{\substack{h=f,r\\k=z,x}}\|\partial_k h\|_{L^\infty(B_\rho\times\mathbb{R}^N\times\mathbb{R})}\int_0^t(|\partial_{\alpha_i}X - e^i| + |\partial_{\alpha_i}z|)ds,\end{aligned}$$

which, after using Grönwall's inequality, gives for all $i = 1,\dots,N$

$$\begin{aligned}&\|\partial_{\alpha_i}X - e^i\|_{L^\infty([0,T]\times\mathbb{R}^N)} + \|\partial_{\alpha_i}z - \partial_{\alpha_i}u_0\|_{L^\infty([0,T]\times\mathbb{R}^N)} \\ \leqslant\;& T\left(\sum_{h=f,r}\|\partial_x h\|_{L^\infty(B_\rho\times\mathbb{R}^N\times\mathbb{R})} + \sum_{h=f,r}\|\partial_z h\|_{L^\infty(B_\rho\times\mathbb{R}^N\times\mathbb{R})}\|\nabla u_0\|_{L^\infty(\mathbb{R}^N)}\right) \\ &\exp\left(T\sum_{\substack{h=f,r\\k=z,x}}\|\partial_k h\|_{L^\infty(B_\rho\times\mathbb{R}^N\times\mathbb{R})}\right).\end{aligned} \tag{35}$$

Now we consider the map

$$H : \mathbb{R}^N \times \mathbb{R} \mapsto \mathbb{R}^N \times \mathbb{R}, \quad (x,t) = H(\alpha,t) := (X(t;\alpha),t)\,.$$

Note that H is C^1 and

$$\nabla_{(\alpha,t)} H(t,\alpha) = \left[\begin{array}{c|c} \nabla_\alpha X & \partial_t X \\ \hline 0 & 1 \end{array}\right],$$

which in view of (35) gives

$$\begin{aligned} \det(\nabla_{(\alpha,t)} H(\alpha,t)) &= \det(\nabla_\alpha X(t,\alpha,t)) = \det(I_N + (\nabla_\alpha X(t,\alpha,t) - I_N)) \\ &\geqslant 1 - P(T), \end{aligned}$$

where $P(T)$ is a homogeneous polynomial of T (its coefficients depend on $\|\partial_k h\|_{L^\infty(B_\rho \times \mathbb{R}^N \times \mathbb{R})}$, $h = f, r,\ k = z, x$). Therefore, given any $\epsilon > 0$, there exists T small enough such that $\det(\nabla_{(\alpha,t)} H(X(t;\alpha),t)) \geqslant \epsilon$, for all $(t,\alpha) \in [0,T] \times \mathbb{R}^N$. From Hadamard theorem [23] it follows that H is invertible in $[0,T] \times \mathbb{R}^N$, and we denote by $H^{-1}(x,t) = (X^{-1}(x,t),t)$ its inverse. Furthermore, we can obtain a bound for $\nabla_{(x,t)} X^{-1}$, which will be used for estimating ∇u, as follows. Note that

$$\nabla_{(x,t)} H^{-1}(x,t) = \left[\begin{array}{c|c} \nabla_x X^{-1} & -\nabla_x X^{-1} \cdot \partial_t X \\ \hline 0 & 1 \end{array}\right].$$

Without loss of generality we assume

$$|||I_N - \nabla_\alpha X|||_{L^\infty([0,T]\times\mathbb{R}^N)} := \max\{N|||I_N - \nabla_\alpha X(t,\alpha)|||,\ (t,\alpha) \in [0,T] \times \mathbb{R}^N\} < 1, \quad (36)$$

which in view of (35) is possible if we take T smaller. Here, for any matrix $A = (a_{i,j}) \in \mathbb{R}^{N\times N}$, we used the sub-multiplicative matrix norm $|||A||| = N \max\{|a_{i,j}|,\ i,j = 1,\dots,N\}$, which satisfies $|||A \cdot B||| \leqslant |||A||| \cdot |||B|||$. Then

$$\begin{aligned} \nabla_x X^{-1}(x,t) &= (\nabla_\alpha X(t,\alpha))^{-1} = (I_N - (I_N - \nabla_\alpha X(t,\alpha)))^{-1}, \\ &= \sum_{n=0}^{\infty} (I_N - \nabla_\alpha X(t,\alpha))^n, \quad x = X(t,\alpha);\ \ \textit{hence} \\ \|\nabla_x X^{-1}(x,t)\|_{L^\infty(\mathbb{R}^N\times[0,T])} &:= \{\max\{\|\partial_{x_i} X_j^{-1}(x,t)\|_{L^\infty(\mathbb{R}^N\times[0,T])},\ \ i,j = 1,\dots,N\} \\ &\leqslant \frac{1}{N(1 - N\|I_N - \nabla_\alpha X\|_{L^\infty([0,T]\times\mathbb{R}^N)})}. \end{aligned} \quad (37)$$

We define $u(x,t) = z(t; X^{-1}(x,t))$, $(x,t) \in \mathbb{R}^N \times [0,T]$. It is classical to show that $u \in C^1(\mathbb{R}^N \times [0,T])$ and it is the unique classical solution of (26).

The estimate (29) follows from (32). The estimate for ∇u in (30) follows from $\partial_{x_i} u = \sum_{j=1}^N \partial_{\alpha_j} z \partial_{x_i} \alpha_j$, in combination with (37) and (35). The estimate for $\partial_t u$ in (30) follows from the estimate for ∇u and by using Equation (26). □

Proof of Theorem 2.1.
(a) We apply Theorem 2.2 with $u_0^\pm$ instead of u_0, and obtain two $C^1(\mathbb{R}^N\times[0,T])\cap W^{1,\infty}(\mathbb{R}^N\times[0,T])$ solutions $u^\pm$. The estimates (24) and (25) follow from (29) and (30).
(b) With $u^\pm$ known, we consider the problem (9), with a and b as given in (12). We note that (9) is of the form (26), in $N-1$ dimensions instead of N. Furthermore, a and b are C^1 functions because u^-, u^+ and g_0 are C^1. Then (9) has a unique solution $g\in C^1(\mathbb{R}^{N-1}\times\times[0,T])\cap W^{1,\infty}(\mathbb{R}^{N-1}\times[0,T])$, which shows that $\Gamma=\{(x',g(x',t),t),\ x'\in\mathbb{R}^{N-1},\ t\in[0,T^*]\}$ is a C^1 surface.

To conclude the proof we must show that Γ is indeed the shock wave associated with (2). Indeed, for T small enough, Condition (23) shows that on Γ intersect only pairs of characteristics, one emanating from $\{g_0<x_N\}$ and the other from $\{g_0>x_N\}$, and Γ does not depend on the extensions $u_0^\pm$. □

2.1.3. Leafnet neural network approximation error estimate for scalar shock waves

In this subsection we present an estimate of the error of the approximation of the shock wave Γ by a neural network network with the *leafnet* algorithm presented in Subsection 2.1.1. The following is the error estimate result for the approximation of $u^\pm$.

Theorem 2.3. *Let* $\mathbf{u}^\pm$ *be a network minimizer of* $\boldsymbol{\mathcal{L}}^\pm$ *in* $\mathbf{N}_A$, *see* (17) *i.e.*

$$\mathbf{u}^\pm=\operatorname{argmin}\{\boldsymbol{\mathcal{L}}^\pm[\mathbf{u}],\ \ \mathbf{u}\in\mathbf{N}_A\},\tag{38}$$

and consider $(\boldsymbol{\varepsilon}_i^\pm,\boldsymbol{\varepsilon}_b^\pm)\in C^\infty(\mathbb{R}^N\times\mathbb{R})\times C^\infty(\mathbb{R}^N)$ *defined by*

$$\boldsymbol{\varepsilon}_i^\pm=L_i^\pm[\mathbf{u}^\pm],\quad \boldsymbol{\varepsilon}_b^\pm=L_b^\pm[\mathbf{u}^\pm],\tag{39}$$

with $L_i^\pm$ *and* $L_b^\pm$ *given by* (15). *Denoting* $e^\pm=\mathbf{u}^\pm-u^\pm$, *we have*

$$\begin{aligned}\|e^\pm\|_{L^\infty(\mathbb{R}^N\times[0,T])}\ &\leqslant\ (\|\boldsymbol{\varepsilon}_b^\pm\|_{L^\infty(\mathbb{R}^N)}+T\|\boldsymbol{\varepsilon}_i^\pm\|_{L^\infty(\mathbb{R}^N\times[0,T])})\cdot\\ &\quad\cdot e^{T\|f'\|_{L^\infty(B(0,\rho_0^\pm+\boldsymbol{\rho}_0^\pm))}\|\nabla u^\pm\|_{L^\infty(\mathbb{R}^N\times[0,T])}},\end{aligned}\tag{40}$$

where $\rho_0^\pm=\rho_0^\pm(u_0^\pm)$ *and* $\boldsymbol{\rho}_0^\pm=\boldsymbol{\rho}_0^\pm(u_0^\pm,\boldsymbol{\varepsilon}_i^\pm,\boldsymbol{\varepsilon}_b^\pm)$ *are given by* (41).

Proof. First we note that $u^\pm$ solves (10) and $\mathbf{u}^\pm$ solves (39), both problems like (26). Applying (29) and (30) of Theorem 2.2 to $u^\pm$ and $\mathbf{u}^\pm$ gives

$$\|u^\pm\|_{L^\infty(\mathbb{R}^N\times[0,T])}\leqslant\|u_0^\pm\|_{L^\infty(\mathbb{R}^N)}=:\rho_0^\pm,\tag{41a}$$
$$\|\mathbf{u}^\pm\|_{L^\infty(\mathbb{R}^N\times[0,T])}\leqslant\|u_0^\pm\|_{L^\infty(\mathbb{R}^N)}+\|\boldsymbol{\varepsilon}_b^\pm\|_{L^\infty(\mathbb{R}^N)}+T\|\boldsymbol{\varepsilon}_i^\pm\|_{L^\infty(\mathbb{R}^N\times[0,T])}=:\boldsymbol{\rho}_0^\pm\tag{41b}$$

and

$$\|\nabla u^\pm\|_{L^\infty(\mathbb{R}^N\times[0,T])}\leqslant\|\nabla u_0\|_{L^\infty(\mathbb{R}^N)}+C^\pm\|f'\|_{L^\infty(B(0,\rho_0^\pm))}=:\rho_1^\pm,\tag{42a}$$
$$\begin{aligned}\|\nabla\mathbf{u}^\pm\|_{L^\infty(\mathbb{R}^N\times[0,T])}&\leqslant\|\nabla u_0\|_{L^\infty(\mathbb{R}^N)}+\|\nabla_x\boldsymbol{\varepsilon}_b^\pm\|_{L^\infty(\mathbb{R}^N)}\\ &\quad+\mathbf{C}^\pm(\|f'\|_{L^\infty(B(0,\boldsymbol{\rho}_0^\pm))},\|\nabla\boldsymbol{\varepsilon}_i^\pm\|_{L^\infty(\mathbb{R}^N\times[0,T])})=:\boldsymbol{\rho}_1^\pm,\end{aligned}\tag{42b}$$

and the constants $C^{\pm}$, $\mathbf{C}^{\pm}$ are as in Theorem 2.2.

From the definition of $u^{\pm}$ and the assumption (39), $e^{\pm}$ satisfies

$$\begin{aligned}\partial_t e^{\pm} + f(\mathbf{u}^{\pm})\cdot\nabla e^{\pm} &= \boldsymbol{\varepsilon}_i^{\pm}(x,t)\\ &\quad - \left(\int_0^1 f'(u^{\pm}+s(\mathbf{u}^{\pm}-u^{\pm}))\cdot\nabla u^{\pm} ds\right) e^{\pm}, \ \text{ in } \mathbb{R}^N\times(0,T), \qquad (43a)\\ e^{\pm}(\cdot,0) &= \boldsymbol{\varepsilon}_b^{\pm}, \ \text{ on } \mathbb{R}^N. \qquad (43b)\end{aligned}$$

We apply Theorem 2.2 with

$$f(z,x,t)\sim f(\mathbf{u}^{\pm}(x,t)), \quad r(z,x,t)\sim \boldsymbol{\varepsilon}_i^{\pm} - \left(\int_0^1 f'(u^{\pm}+s(\mathbf{u}^{\pm}-u^{\pm}))\cdot\nabla u^{\pm} ds\right) z, \quad u_0\sim \boldsymbol{\varepsilon}_b^{\pm}.$$

From the estimate (29) we get

$$\|e^{\pm}\|_{L^\infty(\mathbb{R}^N\times[0,T])} \ \leqslant \ (\|\boldsymbol{\varepsilon}_b^{\pm}\|_{L^\infty(\mathbb{R}^N)} + T\|\boldsymbol{\varepsilon}_i^{\pm}\|_{L^\infty(\mathbb{R}^N\times[0,T])}) e^{T\|f'\|_{L^\infty(B(0,\rho_0^{\pm}+\boldsymbol{\rho}_0^{\pm}))}\|\nabla u^{\pm}\|_{L^\infty(\Omega\times[0,T])}},$$

which proves the theorem. □

Now we will estimate the error of approximating g by the network $\mathbf{g}$, see Subsection 2.1.1.

Theorem 2.4. *Let $\mathbf{u}^{\pm}$ be a network minimizer of $\boldsymbol{\mathcal{L}}^{\pm}[\mathbf{u}]$ as in Theorem 2.3 and $e^{\pm}=\mathbf{u}^{\pm}-u^{\pm}$, where*

$$\boldsymbol{\varepsilon}_i^{\pm} = L_i^{\pm}[\mathbf{u}^{\pm}], \quad \boldsymbol{\varepsilon}_b^{\pm} = L_b^{\pm}[\mathbf{u}^{\pm}]. \qquad (44)$$

Let g be the unique C^1-solution of (9) satisfying $K[g]=(0,0)$, where $K=(K_i,K_b)$ is defined in (13). Let also $\mathbf{g}$ be a network solution of (21), and let $(\boldsymbol{\sigma}_i^{\pm},\boldsymbol{\sigma}_b^{\pm})\in C^\infty(\mathbb{R}^N\times[0,T])\times C^\infty(\mathbb{R}^N)$ defined by

$$\boldsymbol{\sigma}_i = \mathbf{K}_i[\mathbf{g}], \quad \boldsymbol{\sigma}_b = \mathbf{K}_b[\mathbf{g}], \qquad (45)$$

where $\mathbf{K}$ is defined in (18). Then

$$\begin{aligned}\|\mathbf{g}-g\|_{L^\infty(\mathbb{R}^{N-1}\times[0,T])} \ \leqslant \ &\Bigg(\|\boldsymbol{\sigma}_b\|_{L^\infty(\mathbb{R}^{N-1})} + \|\boldsymbol{\sigma}_i\|_{L^\infty(\mathbb{R}^{N-1}\times[0,T])} +\\ &\sum_{*=\pm} C_0^* T(\|\boldsymbol{\varepsilon}_b^*\|_{L^\infty(\mathbb{R}^N)} + T\|\boldsymbol{\varepsilon}_i^*\|_{L^\infty(\mathbb{R}^N\times[0,T])})\Bigg) e^{C_1 T}, \qquad (46)\end{aligned}$$

where

$$\begin{aligned}C_0^{\pm} &= (1+\|\nabla g\|_{L^\infty(\mathbb{R}^N)})\|f'\|_{L^\infty(B(0,\rho_0^+ +\rho_0^- +\boldsymbol{\rho}_0^+ +\boldsymbol{\rho}_0^-))}\cdot\exp\left(T\rho_1^{\pm}\|f'\|_{L^\infty(B(0,\rho_0^{\pm}+\boldsymbol{\rho}_0^{\pm}))}\right), \qquad (47)\\ C_1 &= (1+\|\nabla g\|_{L^\infty(\mathbb{R}^N)})(\rho_1^+ +\rho_1^-)\|f'\|_{L^\infty(B(0,\rho_0^+ +\rho_0^- +\boldsymbol{\rho}_0^+ +\boldsymbol{\rho}_0^-))}, \qquad (48)\\ \rho_0^{\pm} &= \|u_0^{\pm}\|_{L^\infty(\mathbb{R}^N)}, \qquad (\textit{see } (41)) \qquad (49)\\ \boldsymbol{\rho}_0^{\pm} &= \|u_0^{\pm}\|_{L^\infty(\mathbb{R}^N)} + \|\boldsymbol{\varepsilon}_b^{\pm}\|_{L^\infty(\mathbb{R}^{N-1})} + T\|\boldsymbol{\varepsilon}_i^{\pm}\|_{L^\infty(\mathbb{R}^N\times[0,T])}, \qquad (\textit{see } (41)) \qquad (50)\\ \rho_1^{\pm} &= \|\nabla u_0\|_{L^\infty(\mathbb{R}^N)} + C^{\pm}\|f'\|_{L^\infty(B(0,\rho_0^{\pm}))}. \qquad (\textit{see } (42a)) \qquad (51)\end{aligned}$$

Proof. First, we establish some estimates for g and $\mathbf{g}$ that will be used for the estimate of $\mathbf{g}-g$. In view of (13), (18) and (9), g and $\mathbf{g}$ solve a quasi-linear PDE identical to (26), in dimension $N-1$ instead of N. Identifying for each of equations (13), (18), the functions f and r and u_0 (with reference to (26)) and applying the estimate (29) to g and $\mathbf{g}$ gives

$$\begin{aligned}\|g\|_{L^\infty(\Omega'\times[0,T])} &\leqslant \|g_0\|_{L^\infty(\mathbb{R}^{N-1})} + T\|f_N\|_{L^\infty(B(0,\rho_0^-+\rho_0^+))} =: \rho, &(52)\\ \|\mathbf{g}\|_{L^\infty(\Omega'\times[0,T])} &\leqslant \|g_0\|_{L^\infty(\mathbb{R}^{N-1})} + \|\boldsymbol{\sigma}_b\|_{L^\infty(\mathbb{R}^{N-1})}\\ &+ T(\|f_N\|_{L^\infty(B(0,\boldsymbol{\rho}_0^-+\boldsymbol{\rho}_0^+))} + \|\boldsymbol{\sigma}_i\|_{L^\infty(\mathbb{R}^{N-1}\times[0,T])}) =: \boldsymbol{\rho}. &(53)\end{aligned}$$

Next, we estimate $e := \mathbf{g}-g$. For this, we show first that e satisfies a first order PDE like (26). Indeed, subtracting $K[g]=0$ from $\mathbf{K}[\mathbf{g}] = (\boldsymbol{\sigma}_i, \boldsymbol{\sigma}_b)$ we get

$$\begin{aligned}\partial_t e + \mathbf{a}(\mathbf{g},x',t)\cdot\nabla e &= \boldsymbol{\sigma}_i + (\mathbf{b}(\mathbf{g},x',t) - b(g,x',t)) - (\mathbf{a}(\mathbf{g},x',t) - a(g,x',t))\cdot\nabla g\\ &=: \boldsymbol{\sigma}_i + r_b(e,x',t) - r_a(e,x',t) \quad in\ \mathbb{R}^{N-1}\times[0,T], &(54a)\\ e(\cdot,0) &= \boldsymbol{\sigma}_b \quad on\ \mathbb{R}^{N-1}. &(54b)\end{aligned}$$

We note that the equation (54) is similar (26), and e is a C^1-solution of (54). Furthermore, we note that the fluxes $\mathbf{a}(\mathbf{g},x',t)$ in (54a) are C^1, but the right hand side of (54a) is only C^0 which is due to the presence of ∇g. However, due to the fact that $\mathbf{a}(\mathbf{g},x',t)$ satisfies the assumptions (27), the estimate (29) applies.

We will rewrite r_a and r_b as functions of the form $k(x',t)e$. For this set

$$\ell(s,u,v) = su + (1-s)v, \quad e^\pm = \mathbf{u}^\pm - u^\pm.$$

Then r_b can be written as follows

$$\begin{aligned}r_b(e,x',t) &= \mathbf{b}(\mathbf{g}(x',t),x',t) - b(\mathbf{g}(x',t),x',t) + b(\mathbf{g}(x',t),x',t) - b(g(x',t),x',t)\\ &= \int_0^1 (f_N(\ell(s,\mathbf{u}^+(x',\mathbf{g},t),\mathbf{u}^-(x',\mathbf{g},t))) - f_N(\ell(s,u^+(x',\mathbf{g},t),u^-(x',\mathbf{g},t))))ds\\ &+ \int_0^1 (f_N(\ell(s,u^+(x',\mathbf{g},t),u^-(x',\mathbf{g},t))) - f_N(\ell(s,u^+(x',g,t),u^-(x',g,t))))ds\\ &= \int_0^1\int_0^1 f_N'(\ell(\alpha,\ell(s,\mathbf{u}^+(x',\mathbf{g},t),\mathbf{u}^-(x',\mathbf{g},t)),\ell(s,u^+(x',\mathbf{g},t),u^-(x',\mathbf{g},t)))d\alpha\\ &\quad\times\ell(s,e^+(x',\mathbf{g},t),e^-(x',\mathbf{g},t))ds\\ &+ \int_0^1\int_0^1 f_N'(\ell(\alpha,\ell(s,u^+(x',\mathbf{g},t),u^-(x',\mathbf{g},t)),\ell(s,u^+(x',g,t),u^-(x',g,t)))d\alpha\\ &\quad\times\ell\bigg(s,\int_0^1\partial_{x_N}u^+(x',\ell(\alpha,\mathbf{g},g),t)))d\alpha,\int_0^1\partial_{x_N}u^-(x',\ell(\alpha,\mathbf{g},g),t)))d\alpha\bigg)ds\\ &\quad\times e, &(55)\end{aligned}$$

which shows that r_b satisfies (28) (without the term $\nabla_{(z,x)}$) with

$$\|r_b(0,\cdot,\cdot)\|_{L^\infty(\mathbb{R}^{N-1}\times[0,T])} \leqslant \|f_N'\|_{L^\infty(B(0,\rho_0^++\rho_0^-+\boldsymbol{\rho}_0^++\boldsymbol{\rho}_0^-))}\sum_{*=\pm}\|e^+\|_{L^\infty(\mathbb{R}^N\times[0,T])}, \tag{56a}$$

$$C_r^b \leqslant \|f_N'\|_{L^\infty(B(0,2(\rho_0^+ + \rho_0^-)))} \sum_{*=\pm} \|\nabla u^*\|_{L^\infty(\mathbb{R}^N \times [0,T])}. \tag{56b}$$

Similarly, for r_a we have

$$\begin{aligned}
r_a(e,x',t) &= \Big((\mathbf{a}(\mathbf{g}(x',t),x',t) - a(\mathbf{g}(x',t),x',t)) + (a(\mathbf{g}(x',t),x',t) - a(g(x',t),x',t)) \Big) \cdot \nabla g \\
&= \int_0^1 \int_0^1 f_i'(\ell(\alpha, \ell(s, \mathbf{u}^+(x',\mathbf{g},t), \mathbf{u}^-(x',\mathbf{g},t)), \ell(s, u^+(x',\mathbf{g},t), u^-(x',\mathbf{g},t)))d\alpha \\
&\qquad \times \ell(s, e^+(x',\mathbf{g},t), e^-(x',\mathbf{g},t)) \partial_i g ds \\
&+ \int_0^1 \int_0^1 f_i'(\ell(\alpha, \ell(s, u^+(x',\mathbf{g},t), u^-(x',\mathbf{g},t)), \ell(s, u^+(x',g,t), u^-(x',g,t)))d\alpha \\
&\qquad \times \ell\Big(s, \int_0^1 \partial_{x_N} u^+(x', \ell(\alpha,\mathbf{g},g),t)))d\alpha, \int_0^1 \partial_{x_N} u^-(x', \ell(\alpha,\mathbf{g},g),t)))d\alpha \Big) ds \\
&\qquad \times \partial_i g e \,.
\end{aligned} \tag{57}$$

Like for r_b, it implies that r_a satisfies (28) (without the term with $\nabla_{(z,x)}$, which is not needed for the estimate (29)) with

$$\begin{aligned}
&\|r_a(0,\cdot,\cdot)\|_{L^\infty(\mathbb{R}^{N-1}\times[0,T])} \\
&\qquad \leqslant \|f'\|_{L^\infty(B(0,\rho_0^+ + \rho_0^- + \boldsymbol{\rho}_0^+ + \boldsymbol{\rho}_0^-))} \|\nabla g\|_{L^\infty(\mathbb{R}^N)} \sum_{*=\pm} \|e^*\|_{L^\infty(\mathbb{R}^N\times[0,T])}, \qquad (58a) \\
&\qquad C_r^a \leqslant \|f_N'\|_{L^\infty(B(0,2(\rho_0^+ + \rho_0^-)))} \|\nabla g\|_{L^\infty(\mathbb{R}^N)} \sum_{*=\pm} \|\nabla u^*\|_{L^\infty(\mathbb{R}^N\times[0,T])}. \qquad (58b)
\end{aligned}$$

Then applying (29) to (54), and using (58) and (56) gives

$$\begin{aligned}
|e| &\leqslant (\|\boldsymbol{\sigma}_b\|_{L^\infty(\mathbb{R}^{N-1})} + \|\boldsymbol{\sigma}_i\|_{L^\infty(\mathbb{R}^{N-1}\times[0,T])} + T\|r_a(0,\cdot) + r_b(0,\cdot)\|_{L^\infty(\mathbb{R}^{N-1}\times[0,T])}) e^{T(C_r^a + C_r^b)} \\
&\leqslant \big(\|\boldsymbol{\sigma}_b\|_{L^\infty(\mathbb{R}^{N-1})} + \|\boldsymbol{\sigma}_i\|_{L^\infty(\mathbb{R}^{N-1}\times[0,T])} + \\
&\qquad T(1 + \|\nabla g\|_{L^\infty(\mathbb{R}^N)}) \|f'\|_{L^\infty(B(0,\rho_0^+ + \rho_0^- + \boldsymbol{\rho}_0^+ + \boldsymbol{\rho}_0^-))} \big) \sum_{*=\pm} \|e^*\|_{L^\infty(\mathbb{R}^N\times[0,T])} \big) \\
&\qquad \times \exp\Big(T(1 + \|\nabla g\|_{L^\infty(\mathbb{R}^N)}) \|f'\|_{L^\infty(B(0,\rho_0^+ + \rho_0^- + \boldsymbol{\rho}_0^+ + \boldsymbol{\rho}_0^-))} \sum_{*=\pm} \|\nabla u^*\|_{L^\infty(\mathbb{R}^N\times[0,T])} \Big),
\end{aligned}$$

which after replacing $\|e^\pm\|_{L^\infty(\mathbb{R}^N)}$ as given by (40) and $\|\nabla u^\pm\|_{L^\infty(\mathbb{R}^N)}$ as given by (42) completes the proof. □

Remark 2.1. *The estimate* (46) *provides a bound for the error* $\|\mathbf{g} - g\|_{L^\infty(\mathbb{R}^{N-1}\times[0,T])}$ *in terms of data of the problem (*u_0 *and* f*), and the network approximation residuals* $\boldsymbol{\sigma}_i$*,* $\boldsymbol{\sigma}_b$*, and* $\boldsymbol{\varepsilon}_i^\pm$*,* $\boldsymbol{\varepsilon}_b^\pm$*, in the whole space. The estimate* (46) *still provides an estimate for* $\mathbf{g} - g$ *in the case, for example* $\Omega' = \prod_{i=1}^{N-1}(a_i, b_i)$ *and* $\Omega = \Omega' \times (a_N, b_N)$*, as demonstrated by the following arguments.*
(i) We consider the extensions $u_0^\pm$ *with compact support, and therefore the solution* $u^\pm$ *will*

have support in $D \times [0,T]$, *with appropriate* $D = \prod_{i=1}^{N}(c_i, d_i)$, *and* g *will have support in* $D' = \prod_{i=1}^{N-1}(c_i, d_i)$. *As we can always arrange to have the neural networks functions identically zero outside* $D \times [0,T]$, *the estimate* (46) *is reduced in* $D' \times [0,T]$.
(ii) We will assume that the Sobolev index number k, m, p, q *in the leafnet algorithm in Subsection 2.1.1 are such that the embeddings* $H^m(D) \subset L^\infty(D)$, $H^k(D \times [0,T]) \subset L^\infty(D \times [0,T])$, $H^q(D') \subset L^\infty(D')$, $H^p(D' \times [0,T]) \subset L^\infty(D' \times [0,T])$ *are continuous. Then the estimate* (46) *provides a bound of* $\|\mathbf{g} - g\|_{L^\infty(D' \times [0,T])}$ *of the form*

$$\begin{aligned} \|\mathbf{g} - g\|_{L^\infty(D' \times [0,T])} \leqslant & \left(\|\boldsymbol{\sigma}_b\|_{H^m(\mathbb{R}^{N-1})} + \|\boldsymbol{\sigma}_i\|_{H^k(D' \times [0,T])} + \right. \\ & \left. \sum_{*=\pm} D_0^* T(\|\boldsymbol{\varepsilon}_b^*\|_{H^q(D)} + T\|\boldsymbol{\varepsilon}_i^*\|_{H^p(D \times [0,T])}) \right) e^{D_1 T}, \end{aligned} \tag{59}$$

with appropriate constants D_0^*, D_1. *The estimate* (59) *states that the* L^∞ *error of shock approximation is bounded by the Sobolev norm of network residuals* $\boldsymbol{\sigma}_i$, $\boldsymbol{\sigma}_b$, *and* $\boldsymbol{\varepsilon}_i^\pm$, $\boldsymbol{\varepsilon}_b^\pm$.

2.2. Shock wave formation

The method presented in Section 2.1 can be used to handle the shock formation. Namely, we assume that at the initial time there is no shocks, and at a certain time $t_* \in (0,T)$ a shock appears, which develops until the time T. We assume that there is only one shock which is formed, and we denoted by Γ the corresponding discontinuity surface.

Let $t_1 \in (t_*, T)$ and set $\Gamma_1 = \Gamma \cap \{t = t_1\}$. We assume that Γ_1 is the graph of a C^1-function g_1, i.e. $\Gamma_1 = \{(x', g_1(x')),\ x' \in \Omega'\}$. If $u_1(x) := u(x, t_1)$ were known, then we could proceed as in Section 2.1 to track the shock, with u_1 instead of u_0 and Γ_1 instead of Γ_0.

Even though u_1 and Γ_1 are not known, we can still proceed as in Subsection 2.1 as follows. We consider a virtual discontinuity curve Γ_0 with equations $x_N = g_0(x')$, which separates Ω in $\Omega^\pm$, and generates two initial functions $u_0^\pm$ as in (4). Of course $u_0^- = u_0^+$ on Γ_0. Next, we solve $u^\pm$ and g as in (10) and (9). The time formation t_* is smallest time when $[\![u]\!]$ on Γ becomes discontinuous. Therefore, the solution $u(\cdot, t)$ is continuous for $t \in [0, t_*]$, any g will solve (9). However, for $t \in (t_*, T)$, g must satisfy the Rankine-Hugoniot condition and therefore g will parameterize the shock surface Γ. We illustrate this method with numerical results in Section 4.

2.3. Neural network solution for rarefaction waves

In this section we assume that u_0 has only one discontinuity $\Gamma_0 = \{(x', g_0(x'),\ x' \in \Omega'\}$, which develops a rarefaction wave. We can apply the classical PINN methods, which provide a smooth approximation of u with neural networks. As u is discontinuous along Γ_0, this network will then not approximate accurately the solution near Γ_0.

2.3.1. Discontinuous rarefaction waves

We can approximate sharply the solution even near the discontinuity curve Γ_0 by using a kind of self similar solution as follows. As the discontinuity on Γ_0 develops a rarefaction

wave, there exists a domain $Q_{\mathrm{rw}} \subset Q$, bounded by the characteristic surfaces $\Gamma^{\pm}$, made of characteristic curves starting on Γ_0 given by

$$\begin{aligned}\Gamma^{\pm} \;=\; \{(x', x_N, t),\;\; & x_i = \alpha_i + t f_i(u_0^{\pm}(\alpha, g_0(\alpha))),\;\; (\alpha, t) \in Q',\;\; i = 1, \ldots, N-1, \\ & x_N = g_0(\alpha) + t f_N(u_0^{\pm}(\alpha, g_0(\alpha)))\}\,. \end{aligned} \tag{60}$$

At least for small T, the map $(\alpha, t) \in Q' \mapsto (x', t)$, $x' = (x_1, \ldots, x_{N-1})$, is invertible. Then (60) provides $x_N = g^{\pm}(x', t)$. Depending on f and $u_0^{\pm}$, one may solve $g^{\pm}$ analytically. Alternatively, $g^{\pm}$ can be evaluated numerically, see Subsection 2.3.2. In the following, we assume that $g^{\pm}$ are known, and therefore $\Gamma^{\pm}$ is the graph of the function $x_N = g^{\pm}(x', t)$. Hence, we have

$$Q_{\mathrm{rw}} = \{(x', x_N, t),\;\; x_N \in (g^-(x', t), g^+(x', t)),\; (x', t) \in Q'\}. \tag{61}$$

The solution u satisfies

$$\partial_t u + f(u) \cdot \nabla u = 0, \quad \text{in } Q_{\mathrm{rw}}, \tag{62a}$$

$$u = u^- \;\text{ on }\; \Gamma^-, \tag{62b}$$

$$u = u^+ \;\text{ on }\; \Gamma^+. \tag{62c}$$

As the solution u is discontinuous on Γ_0, to approximate it accurately with neural networks, we consider the change of variables

$$(x', x_N, t) \in Q \mapsto (x', s_N, t) \in \Omega' \times (0,1) \times (0, T), \quad s_N = \frac{x_N - g^-(x', t)}{g^+(x', t) - g^-(x', t)},$$

with corresponding inverse

$$(x', s_N, t) \mapsto (x', x_N, t), \quad x_N = g^-(x', t) + s_N(g^+(x', t) - g^-(x', t)),$$

and define w by

$$w(x', s_N, t) = u(x', x_N, t). \tag{63}$$

We can write (62) in terms of the function w. For this we evaluate

$$\begin{aligned}\partial_t u(x', x_N, t) \;&=\; \partial_t w(x', s_N, t) \\ &-\; \partial_{s_N} w(x', s_N, t)\bigg(\frac{\partial_t g^-(x', t)}{g^+(x', t) - g^-(x', t)} + \\ &\qquad \frac{x_N - g^-(x', t)}{(g^+(x', t) - g^-(x', t))^2}\partial_t(g^+(x', t) - g^-(x', t))\bigg), \\ \partial_{x_i} u(x', x_N, t) \;&=\; \partial_{x_i} w(x', s_N, t) \\ &-\; \partial_{s_N} w(x', s_N, t)\bigg(\frac{\partial_{x_i} g^-(x', t)}{g^+(x', t) - g^-(x', t)} +\end{aligned}$$

$$\frac{x_N - g^-(x',t)}{(g^+(x',t)-g^-(x',t))^2}\partial_{x_i}(g^+(x',t)-g^-(x',t))\Bigg),$$

$$\partial_{x_N}u(x',x_N,t) \;=\; \partial_{s_N}w(x',s_N,t)\frac{1}{g^+(x',t)-g^-(x',t)},$$

for $i = 1,\ldots,N-1$. Combining these relations with (62) gives for $w = w(x',s_N,t)$:

$$\partial_t w + \sum_{i=1}^{N-1} h_i\partial_{x_i}w + h_N\partial_{s_N}w = 0, \;\; \text{in } \{(x',s_N,t)\in\Omega'\times(0,1)\times(0,T)\}, \tag{64a}$$

$$w(x',0,t) = u^-(x',g^-(x',t),t), \;\; \text{on } \Omega'\times(0,T), \tag{64b}$$

$$w(x',1,t) = u^+(x',g^+(x',t),t), \;\; \text{on } \Omega'\times(0,T), \tag{64c}$$

$$w(x',s_N,0) = w_0(x',s_N), \;\; \text{on } \Omega'\times(0,1), \tag{64d}$$

with w_0 any smooth function satisfying

$$w_0(x',0) = u^-(x',g_0(x')), \quad w_0(x',1) = u^+(x',g_0(x')), \quad x'\in\Omega', \tag{65}$$

for example

$$w_0(x',s_N) = (1-s_N)u_0^-(x',g_0(x')) + s_N u_0^+(x',g_0(x')),$$

and

$$h_i(w,x',s_N,t) \;=\; f_i(w), \tag{66}$$

$$\begin{aligned} h_N(w,x',s_N,t) \;=\; & -\Bigg(\frac{\partial_t g^-(x',t)}{g^+(x',t)-g^-(x',t)} + \\ & \frac{x_N(x',s_N,t)-g^-(x',t)}{(g^+(x',t)-g^-(x',t))^2}\partial_t(g^+(x',t)-g^-(x',t))\Bigg) \\ & -f_i(w)\Bigg(\frac{\partial_{x_i}g^-(x',t)}{g^+(x',t)-g^-(x',t)} + \\ & \frac{x_N(x',s_N,t)-g^-(x',t)}{(g^+(x',t)-g^-(x',t))^2}\partial_{x_i}(g^+(x',t)-g^-(x',t))\Bigg) \\ & -f_N(w)\frac{1}{g^+(x',t)-g^-(x',t)}. \end{aligned} \tag{67}$$

We can solve w in (64) and then the solution u of (62) is given by

$$u(x',x_N,t) = \begin{cases} u^-(x',x_N,t), & x_N < g^-(x',t), \\ w\left(x',\dfrac{x_N-g^-(x',t)}{g^+(x',t)-g^-(x',t)},t\right), & g^-(x',t) < x_N < g^+(x',t), \\ u^+(x',x_N,t), & x_N > g^+(x',t)\,. \end{cases} \tag{68}$$

2.3.2. Neural network algorithm for rarefaction waves

We implement the method presented in Subsection 2.3 with neural networks. The method yet again benefits from the accurate approximation of rarefaction waves avoiding spurious regularization. More specifically, we introduce two $N+1$-dimensional neural networks $\mathbf{u}^{\pm}(x', x_N, t)$ for solving (10). In addition we introduce two N-dimensional neural networks $\mathbf{g}^{\pm}$ for describing the curved faces $\Gamma^{\pm}$ of Q_{rw} as defined in (60), and one $N+1$-dimensional neural network $\mathbf{w}(x', s_N, t)$ for solving (64). Then we proceed as follows

1. *Find approximate solutions* $\mathbf{u}^{\pm}$ *to IVP* (10)*, by solving* (17).
2. If possible, solve analytically $x_N = g^{\pm}(x', t)$ as given by (60), i.e.

$$x_i = \alpha_i + t f_i(u_0^{\pm}(\alpha, g_0(\alpha))), \quad i = 1, \dots, N-1, \quad \alpha = (\alpha_1, \dots, \alpha_{N-1}), \tag{69a}$$

$$x_N = g_0(\alpha) + t f_N(u_0^{\pm}(\alpha, g_0(\alpha))). \tag{69b}$$

If not, we search for a neural network approximation of $x_N = g^{\pm}(x', t)$. In this goal, we first introduce the following operator

$$G^{\pm}[g] \;=\; g_0(\alpha) + t f_N(u_0^{\pm}(\alpha, g_0(\alpha))) - g\Big(\prod_{i=1}^{N-1}(\alpha_i + t f_i(u_0^{\pm}(\alpha, g_0(\alpha)))), t\Big). \tag{70}$$

Then, given an architecture A in N dimensions, with parameters Θ_A and corresponding set of admissible functions $\mathbf{N}_A$, and for $k \in \mathbb{N}_0$ and $\mathbf{g} \in \mathbf{N}_A$ we define

$$\mathcal{G}^{\pm}[\mathbf{g}] = \sum_{\substack{\alpha \in \mathbb{N}_0^{N-1},\ \beta \in \mathbb{N}_0 \\ |\alpha| + \beta \leqslant k}} \int_0^T \int_{\Omega'} |D_{x'}^{\alpha} D_t^{\beta} G_i^{\pm}[\mathbf{g}]|^2 dx' dt\,. \tag{71a}$$

We then look for $\mathbf{g}^{\pm}$ defined by

$$\mathbf{g}^{\pm} = \operatorname{argmin}\{\mathcal{G}^{\pm}[\mathbf{g}],\ \ \mathbf{g} \in \mathbf{N}_A\}, \tag{72}$$

hence providing the network approximations $\mathbf{g}^{\pm}$ of $g^{\pm}$. In the following we assume that $g^{\pm}$ or $\mathbf{g}^{\pm}$ are known, and for simplicity they are denoted by $g^{\pm}$.

3. *Solution to the IVP* (64). Motivated by (64) we introduce the operator R as follows

$$R[w] = (R_i[w], R_{\pm}[w], R_b[w]), \tag{73a}$$

$$R_i[w] = \partial_t w + \sum_{i=1}^{N-1} h_i \partial_{x_i} w + h_N \partial_{s_N} w, \tag{73b}$$

$$R_{\pm}[w] = (w(x', 0, t) - u^{-}(x', g^{-}(x', t), t), w(x', 1, t) - u^{+}(x', g^{+}(x', t), t)), \tag{73c}$$

$$R_b[w] = w(x', s_N, 0) - w_0(x', s_N), \tag{73d}$$

and for a given architecture A in $N+1$ dimensions, with feasible network space $\mathbf{N}_A$, consider the operator $\mathcal{R}[\mathbf{v}]$, $\mathbf{v} \in \mathbf{N}_A$, given by

$$\mathcal{R}[\mathbf{v}] = \sum_{\substack{\alpha \in \mathbb{N}_0^N, \beta \in \mathbb{N}_0 \\ |\alpha| + \beta \leqslant k}} \int_0^T \int_0^1 \int_{\Omega'} |D_{x'}^{\alpha'} D_{s_N}^{\alpha_N} D_t^{\beta} R_i[\mathbf{v}]|^2 dx' ds_N dt \tag{74a}$$

$$+ \sum_{\substack{\alpha' \in \mathbb{N}_0^{N-1}, \beta \in \mathbb{N}_0 \\ |\alpha'|+\beta \leqslant m}} \int_0^T \int_{\Omega'} |D_{x'}^{\alpha'} D_t^{\beta} R_{\pm}[\mathbf{v}]|^2 dx' dt \tag{74b}$$

$$+ \sum_{\substack{\alpha \in \mathbb{N}_0^N \\ |\alpha| \leqslant m}} \int_0^1 \int_{\Omega'} |D_{x'}^{\alpha'} D_{s_N}^{\alpha_N} R_b^2[\mathbf{v}]|^2 dx' ds_N, \tag{74c}$$

where $\alpha = (\alpha', \alpha_N)$. Then $\mathbf{w} \in \mathbf{N}_A$ is defined by

$$\mathbf{w} = \operatorname{argmin}\{\mathcal{R}[\mathbf{v}], \quad \mathbf{v} \in \mathbf{N}_A\}. \tag{75}$$

4. *Reconstruction of the approximate solution* $\mathbf{u}$ *to* (2)*, for* $(x', x_N, t) \in Q$

$$\mathbf{u}(x', x_N, t) = \begin{cases} \mathbf{u}^-(x', x_N, , t), & \text{if } x_N < g^-(x', t)\,, \\ \mathbf{w}\left(x', \dfrac{x_N - g^-(x', t)}{g^+(x', t) - g^-(x', t)}, t\right), & \text{if } x_N \in (g^-(x', t), g^+(x', t)), \\ \mathbf{u}^+(x', x_N, t), & \text{if } x_N > g^+(x', t)\,. \end{cases}$$

3. Systems of hyperbolic conservation laws in N dimensions

In this section we provide a short discussion on the extension of the methods we developed in Section 2 for hyperbolic scalar conservation laws to N dimensional hyperbolic systems of conservation laws, given by

$$\begin{aligned} \partial_t u + \nabla \cdot F(u) : &= \partial_t u + \sum_{k=1}^N \partial_k (F_k(u)) \\ &= \partial_t u + \sum_{k=1}^N \nabla F_k(u) \cdot \partial_k u = 0 \qquad \text{in } Q = \Omega \times (0, T), \end{aligned} \tag{76a}$$

$$u(\cdot, 0) = u^0(\cdot), \qquad \text{on } \Omega, \tag{76b}$$

where $\Omega \subset \mathbb{R}^N$ is open, $u : (x, t) = (x_1, \ldots, x_N, t) \in Q \mapsto u(x, t) = (u_i(x, t)) \in \mathbb{R}^m$, $m \geqslant 2$, $\partial_t u = (\partial_t u_i) \in \mathbb{R}^m$, $\partial_k u = (\partial_k u_i) \in \mathbb{R}^m$, $F(u) = (F_1(u), \ldots, F_N(u)) : \mathbb{R}^m \mapsto \mathbb{R}^{m \times N}$, with $F_k = (F_k^i) \in \mathbb{R}^m$, $\nabla F_k = (\partial_j F_k^i) \in \mathbb{R}^{m \times N}$, $F_k^i \in C^2(\mathbb{R}; \mathbb{R})$, $i, j = 1, \ldots, m$, $k = 1, \ldots, N$.

Moreover, we assume that $u^0 \in L^1(\Omega; \mathbb{R}^m) \cap TV(\Omega; \mathbb{R}^m)$. We again impose incoming flux boundary condition $\partial\Omega$. We refer to [19] for a full description of the incoming flux conditions for systems.

3.1. Shock waves for the systems of hyperbolic conservation laws

The leafnet algorithm we have presented in Subsection 2.1 can be applied for systems, with minimal changes. Indeed, the Rankine-Hugoniot condition (9), which is the key ingredient for tracking the shock wave, in the case of systems is written componentwise as

$$\nu_t [\![u_i]\!] + \sum_{k=1}^N [\![F_k^i(u)]\!] \nu_k \;\; = \;\; 0, \quad i = 1, \ldots, m, \tag{77}$$

where ν is given in (6). As in Subsection 2.1, rearranging (77), we deduce that the function $x_N = g(x', t)$ representing the discontinuity surface Γ satisfies

$$\partial_t g + \frac{[\![F_1^i(u)]\!]}{[\![u_i]\!]}\partial_1 g + \cdots + \frac{[\![F_{N-1}^i(u)]\!]}{[\![u_i]\!]}\partial_{N-1} g =:$$

$$\partial_t g + a_1^i(g, x', t)\partial_1 g + \cdots + a_{N-1}^i(g, x', t)\partial_{N-1} g = r^i(g, x', t) \tag{78a}$$

$$:= \frac{[\![F_N^i(u)]\!]}{[\![u_i]\!]}, \quad i = 1, \ldots, m,$$

$$g(\cdot, 0) = g_0(\cdot)\,. \tag{78b}$$

We assume that F is C^2 and therefore, each equation of (78) has a unique solution, which is g. Therefore, we can find g by solving any of the equations of (78). We note that from numerical viewpoint, to avoid numerical instabilities, among the equations of (78) one would like to choose the one with maximal $[\![u_i]\!]$. In absence of any information about the coefficients a_k^i and r^i, choosing which of the equations of (78) to solve is difficult. In our numerical simulations, we have considered the following equivalent form of (78)

$$[\![u_i]\!]\partial_t g + [\![F_1^i(u)]\!]\partial_1 g + \cdots + [\![F_{N-1}^i(u)]\!]\partial_{N-1} g = [\![F_N^i(u)]\!], \quad i = 1, \ldots, m, \tag{79a}$$

$$g(\cdot, 0) = g_0(\cdot)\,. \tag{79b}$$

We have solved (79) for different i and we have found the same result. We note that other strategies can be used for solving g. One would be to solve the equation obtained by summing all the equations of (79).

Our algorithm for solving g, will be to solve just one of equations of (79). Hence, "solving" the shock Γ of (76) is equivalent to solve the $N-1$ dimensional IVP (78). The leaf method consists in:

1) *Solving (independently) the two IBVP like* (10)*, associated to* (76)*, which provide* u^- *and* u^+ *in* Q.
2) *Solving the Rankine-Hugoniot* (79)*, which provides* g *in* Q'.
3) *Reconstructing the solution from the initial IBVP* (10) *for* $(x', x_N, t) \in Q$*, as follows* (80)

$$u(x', x_N, t) = \begin{cases} u^-(x', x_N, t), & \text{if } x_N < g(x', t)\,, \\ u^+(x', x_N, t), & \text{if } x_N > g(x', t)\,. \end{cases}$$

The neural network implementation of this algorithm is made as in Subsection 2.1.1, but with vector valued neural networks.

3.2. Shock wave formation

We can apply also the method we presented in Subsection 2.2 for finding the time t_* of the shock wave formation and tracking the shock wave.

Indeed, like in Subsection 2.2 for tracking the scalar HCL. We start with a "virtual" discontinuity Γ_0. Then we approximate $u^\pm$, and next the solution g to Rankine-Hugoniout equation (78). The time formation t_* is smallest time when $[\![u]\!]$ on Γ becomes discontinuous.

4. Numerical experiments

In this section, we present a series of numerical experiments illustrating the methods that we have developed in this paper, and applied to two-dimensional scalar shock waves, one- and two-dimensional rarefaction waves, and two-dimensional systems of HCL. As in this section we propose two-dimensional experiments, it is more convenient to present the shocks with functions of the form $x = g(y,t)$ rather than $x_2 = g(x_1,t)$, and consequently, we will use the indices l, r rather than $\pm$.
In all the numerical computations, we use fully connected neural networks with tanh-activation functions, and the loss functions are constructed with H^1-norms. A standard gradient descent algorithm from the library `optax` is used in the training of the networks. In addition, the initial and boundary conditions are directly encoded in the neural networks. Finally, in most of the computations the learning rate is taken equal to 2×10^{-3}.

4.1. Scalar HCL

In this subsection, we propose a numerical illustration of the above strategy applied to scalar HCL for tracking shock waves, rarefaction waves and shock formation, see subsections 2.1, 2.3 and 2.2.

Experiment 1 (2-D shock) Here we consider (2), with $F_1(u) = u^4/4$, $F_2(u) = u^2/4$, $\Omega = (a_1, b_1) \times (a_2, b_2)$, and the following initial condition u_0

$$u_0(x_1,x_2) = \begin{cases} u_0^l(x_1,x_2), & x_1 < g_0(x_2)\,, \\ u_0^r(x_1,x_2), & x_1 > g_0(x_2)\,, \end{cases}$$

where $g_0(x_2) = \beta(x_2 - a_2)^2$ and

$$\begin{aligned} u_0^l(x_1,x_2) &= \gamma(x_1+b_1)\big(1-\tanh[\epsilon(g_0(x_2)-(x_1-(a_1+b_1)/2))]\big) \\ &\quad +\gamma(g_0(x_2)+b)\big(1-\tanh[\epsilon(g_0(x_2)-(x_1-(a_1+b_1)/2))]\big), \\ u_0^r(x_1,x_2) &= 1\,. \end{aligned}$$

and ϵ, β, γ positive constants. Numerically we select $a_2 = b_2 = -a_1 = -b_1 = 2$, $\epsilon = 20$, $\beta = 1/20$, $\gamma = 1/4$, and $T = 1$. In the numerical experiments, the networks for approximating the IBVP on $u^{l,r}$ have 2 hidden layers with 5 neurons, and have 1 hidden layer with 10 neurons for g. The networks are trained with 300 randomly chosen collocation points in x_1, x_2 and t. We report in Fig. 1, the graph of u_0^l (Left) after extension, and the graph of u_0 (Right) in Ω. We then report in Fig. 2, the graph of $u^l(\cdot,\cdot,T)$ (Left) as well as the graph of the reconstructed solution $u(\cdot,\cdot,T)$ (Right). Finally, we report in Fig. 3, the graph of g in Q_2 defining the time-space surface of discontinuity.

For completeness, we also report the graph of convergence of the minimization algorithm in Fig. 3 (Right). This first test illustrates the zero-diffusion approximation of the two-dimensional shock wave.

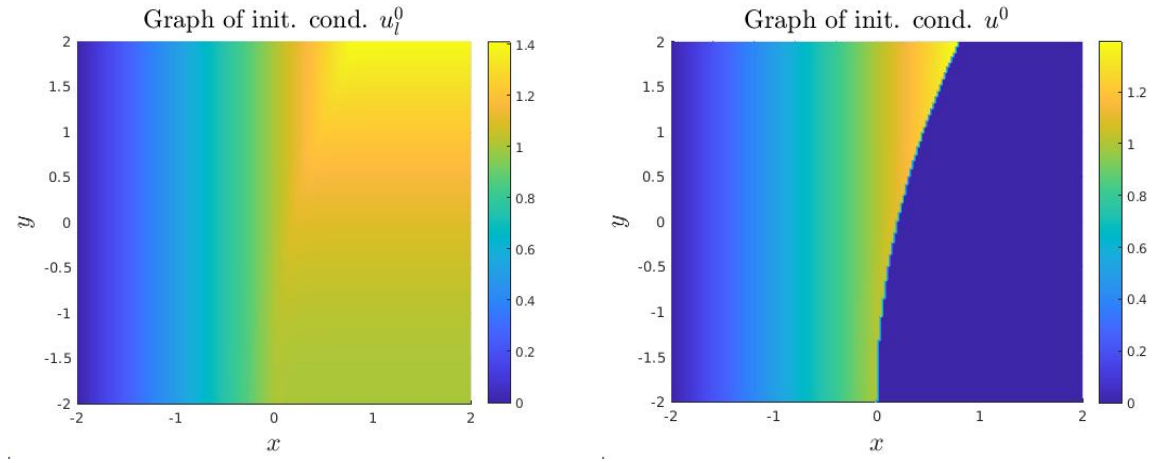

Figure 1: **Experiment 1.** (Left) Graph of u_0^l. (Right) Graph of u_0.

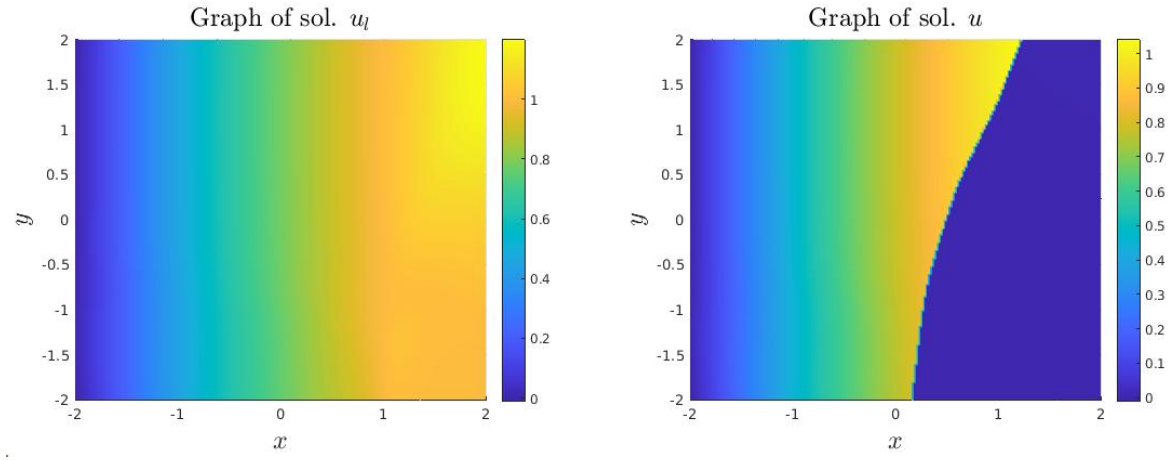

Figure 2: **Experiment 1.** (Left) Solution $\mathbf{u}_l(\cdot,\cdot,T)$. (Right) Solution $\mathbf{u}(\cdot,\cdot,T)$.

Experiment 2 (1-D rarefaction wave) This experiment is devoted to the approximation of one-dimensional rarefaction waves for the following Riemann problem on $Q = \Omega \times [0,T]$, where $\Omega = (-3/2, 3/2) \times (-2, 2)$ and $T = 1.0$.

$$u_0(x_1, x_2) = \begin{cases} u_l, & x_1 < 0, \\ u_r, & x_1 > 0\,, \end{cases}$$

with $u_l = 0$ and $u_r = 1$, and the flux function $F = (F_1, F_2)$ is defined by $F_1(u) = u^2/2$, $F_2(u) = u^2/8$. We implement the method proposed in Subsection 2.3 with 1 layer and 10 neurons in a one-dimensional setting. The networks are trained with 300 randomly chosen collocation points for each variable.

Following the strategy described in Subsection 2.3.1, we simply get (as $u^l = 0$, $u^r = 1$ and $g_0(x_2) = 0$) $g^l(x_2, t) = 0$, $g^r(x_2, t) = t$ and $s_1 = x_1/t$. Hence we recover the standard

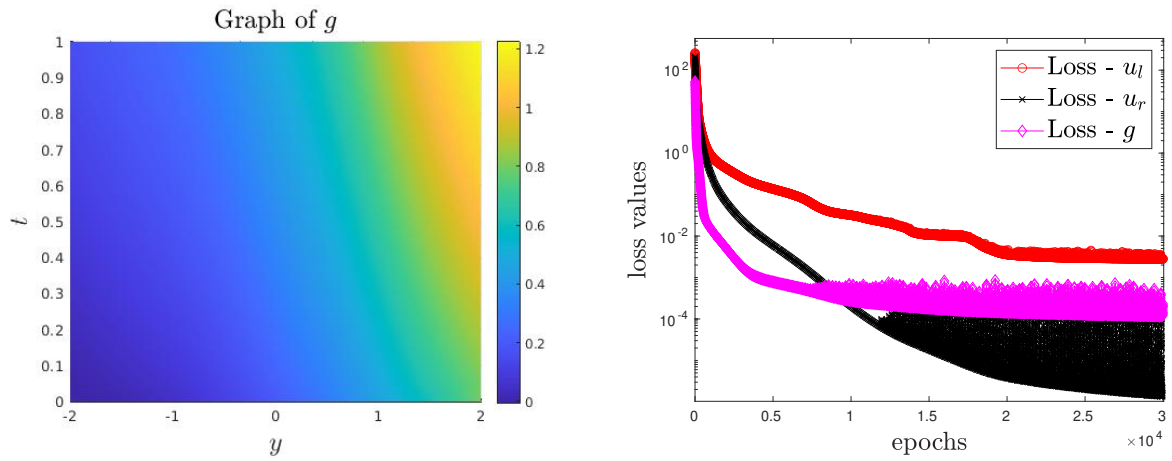

Figure 3: **Experiment 1.** (Left) Graph of $\mathbf{g}$. (Right) Loss functions for training $\mathbf{u}_l$, $\mathbf{u}_r$ and $\mathbf{g}$.

one-dimensional (as the solution is here independent of x_2 and t) rarefaction wave structure $w(s_1, x_2)$.

We then search for neural network $\mathbf{w}$ solution approximate w as proposed in Subsubsection 2.3.2. We report in Fig. 4 the solution $\mathbf{u}$ at $t = 1/4$ (Top-Left) and at $t = T = 1$ (Top-Right). Similarly, we report the corresponding solutions obtained with a direct PINN method and at $t = 1/4$ (Bottom-Left) and $t = 1$ (Bottom-Right). In order to get a fair comparison, we use the same number of layers and neurons in both methods. This example illustrates that the use of self-similar neural networks allows for an accurate approximation of rarefaction waves with artificial regularization of non-smooth regions. We also report in Fig. 5 (Left), the loss functions (constructed using the H^1-norm on $\Lambda_0 = (-2, 2) \times (0, 1)$ with self-similar neural networks) as well as the loss function with the direct PINN-method, to illustrate the convergence of the optimization algorithm.

Experiment 2 bis. (1-D rarefaction wave) In this experiment we consider the same data as in Experiment 2, excepted that we change the flux function $F = (F_1, F_2)$, by $F_1(u) = u^4/4$, $F_2(u) = u^2/8$. We report in Fig. 6 the solution u_{rw} at $t = 1/4$ (Top-Left) and at $t = T = 1$ (Top-Right). Similarly, we report the corresponding solutions obtained with a direct PINN method and at $t = 1/4$ (Bottom-Left) and $t = 1$ (Bottom-Right). We use the same number of layers and neurons in both methods.

For completeness, we also report in Fig. 6 (Right), the loss functions (constructed using the H^1-norm on $\Lambda_0 = (-2, 2) \times (0, 1)$ with self-similar neural networks) as well as the loss function with the direct PINN-method, to illustrate the convergence of the optimization algorithm.

These numerical experiments illustrate the accuracy of the proposed strategy to approximate rarefaction and shock waves.

Experiment 3 (2-D rarefaction wave) This experiment is dedicated to the approximation of 2-D rarefaction wave using the strategy described in Section 2.3, with a neural network

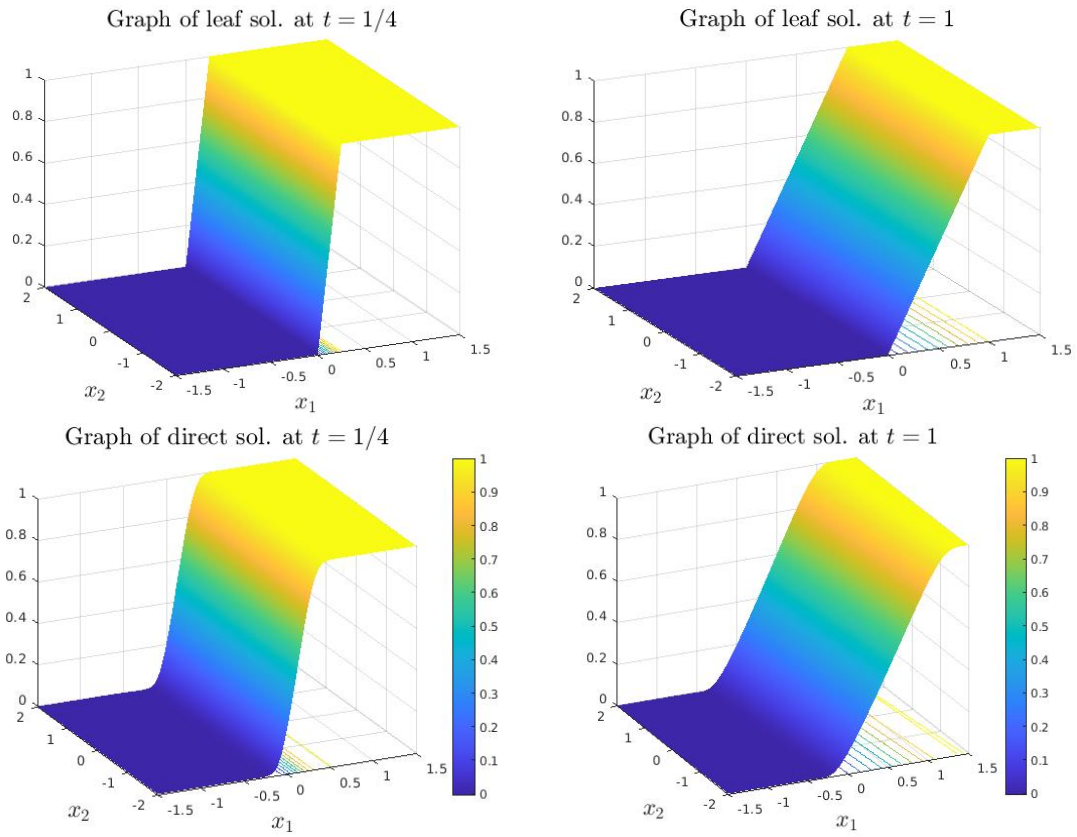

Figure 4: **Experiment 2.** (Top): Solution $\mathbf{u}$ with the method of Subsection 2.3, (Left) at $t = 1/4$, (Right) at $t = 1$. (Bottom): Solution with PINN-method, (Left) at $t = 1/4$, (Right) at $t = 1$.

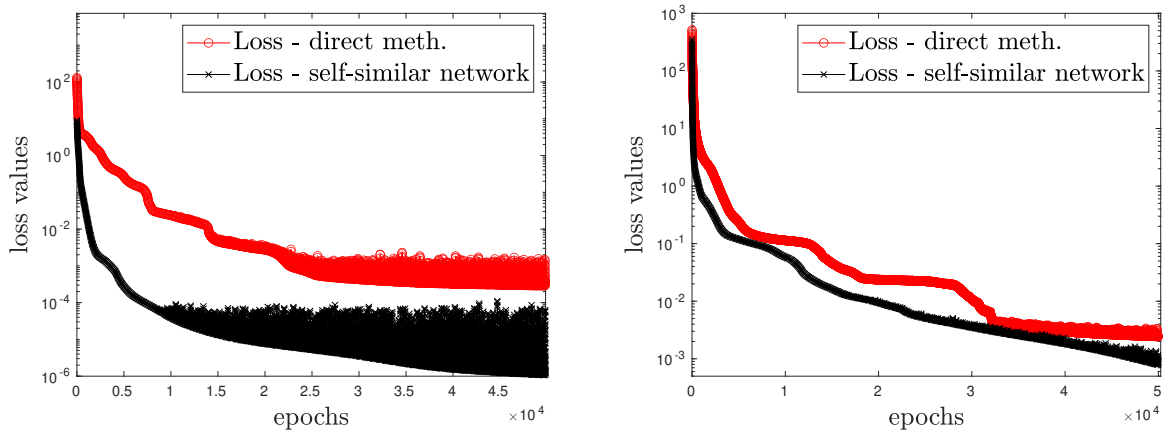

Figure 5: **Experiment 2.** (Left) Loss functions with direct PINN-method and with self-similar networks. **Experiment 2 bis.** (Right) Loss functions with direct PINN-method and with self-similar networks.

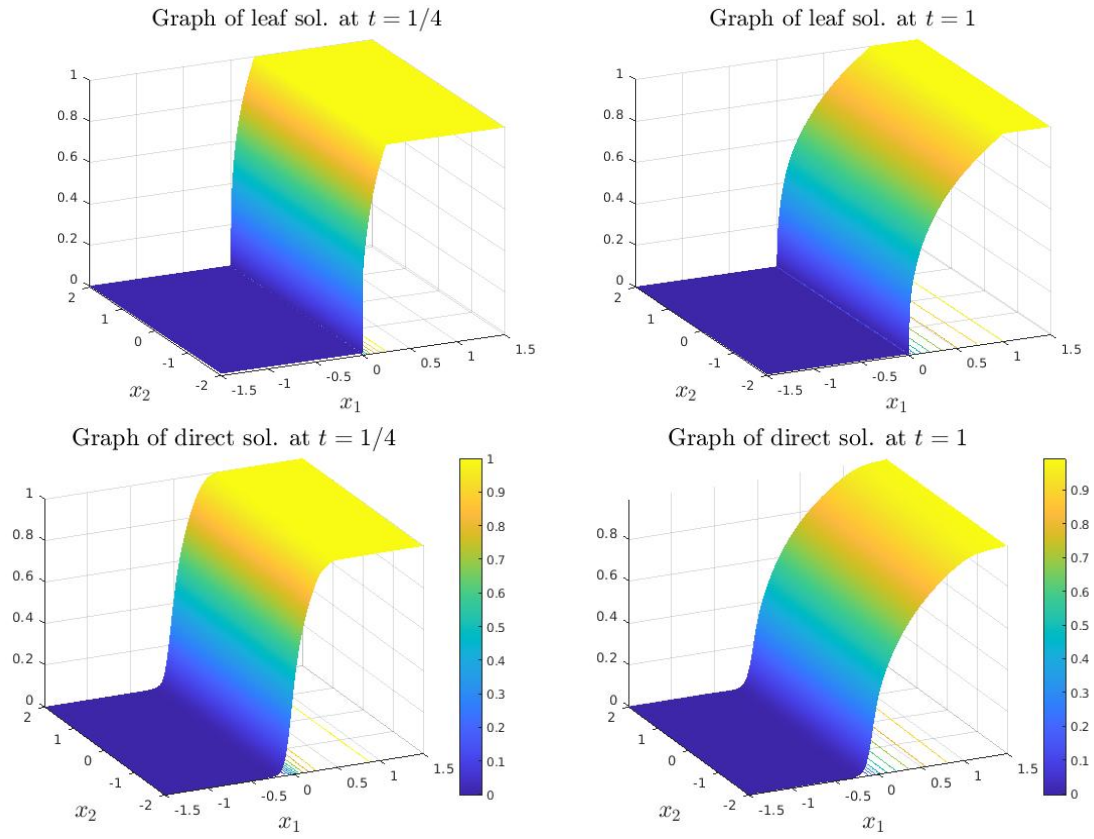

Figure 6: **Experiment 2 bis.** (Top-Left) Solution with self-similar networks at $t = 1/4$. (Top-Right) Solution with self-similar networks at $t = 1$. (Bottom-Left) Solution with PINN-method at $t = 1/4$. (Bottom-Right) Solution with PINN-method at $t = 1$.

having 3 hidden layers and 15 neurons each. The networks are trained with 300 randomly chosen collocation points in x_1, x_2 and t. We consider the following Riemann problem on $Q = \Omega \times [0, T]$, where $\Omega = (-2, 2) \times (-2, 2)$ and $T = 1.1$. The initial data is reported in Fig. 7 (Left)

$$u_0(x_1, x_2) = \begin{cases} u_0^l, & x_1 < g_0(x_2), \\ u_0^r, & x_1 > g_0(x_2), \end{cases}$$

with $u_0^l = 0$ and $u_0^r = 1$, $g_0(x_1, x_2) = (x_2 + 2)^2/10 - 1$, and the flux function $F = (F_1, F_2)$ is defined by $F_1(u) = u^2/2$, $F_2(u) = u^2/8$. We report in Fig. 7 the solution u_{rw} at $t = T = 1.1$ (Right). This example illustrates that the use of self-similar neural networks allows for an accurate approximation of rarefaction waves with no artificial regularization of non-smooth regions.

Experiment 4 (2-D shock formation) This experiment is an illustration of a shock generation from a smooth initial condition. We compare the solution obtained with a direct PINN-method with the algorithm proposed in Subsection 2.2. We consider $Q = \Omega \times [0, T]$,

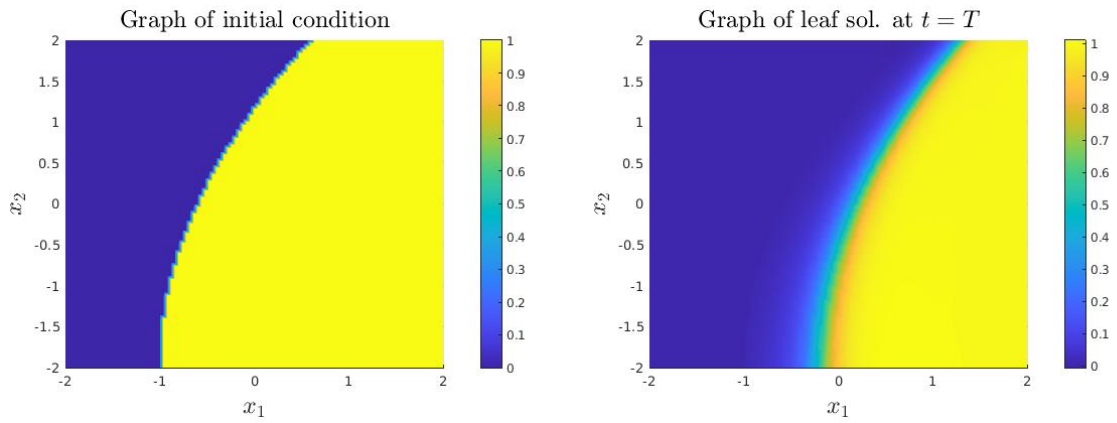

Figure 7: **Experiment 3.** (Left) Initial data. (Right) Solution with self-similar networks at $t = T$.

with $\Omega = (-1, 1) \times (-2, 2)$ and $T = 0.8$. The flux function $F = (F_1, F_2)$ with $F_1(u) = u^2$, $F_2(u) = u^2/4$ and the initial condition is taken as

$$u_0(x_1, x_2) = \frac{1}{2}(1 - \tanh(5(x_1 + (x_2 + 2)^2/20))) .$$

In this example, the solution is smooth up to time $t = t^*$ corresponding to the generation of a shock wave:

$$t^* \quad = \quad -\big(\min_{(x_1^0, x_2^0) \in \Omega} \sum_{i=1}^{2} \partial_{x_i^0}(f_i(u_0(x_1, x_2)))\big)^{-1} , \tag{81}$$

which is equal to 0.2 in the considered example. The algorithm from Subsection 2.2 is applied with 2 hidden layers with 15 neurons each. In Fig. 8 (Left) we plot the initial data u_0 and the solution at $t = t^*$ (Right). In Fig. 9, we report the PINN solution and the leafnet solution (Right) at $t = T$.

We report in Fig. 10 (Left) the surface of discontinuity, and the loss function for the leafnet algorithm in Fig. 10 (Right). This test illustrates the ability of the algorithm to accurately generate and propagate shock waves.

4.2. Systems of HCL

In this subsection, we propose a preliminary illustration of the above strategy applied to a single shock-wave solution to a two-dimensional system. We consider here the two-dimensional shallow-water equations with flat bottom modeling compressible fluid flows, and corresponding to $m = 3$ in (76). The latter models the evolution of a fluid velocity field (u, v) and height h in shallow water. At this stage, the proposed method does not consider initial wave decompositions for instance from an arbitrary discontinuous initial data (or more complex initial data), and is mainly designed for simple waves (rarefaction and/ shock waves. For this reasons we here focus on a relatively simple case (one shock wave). The conservative

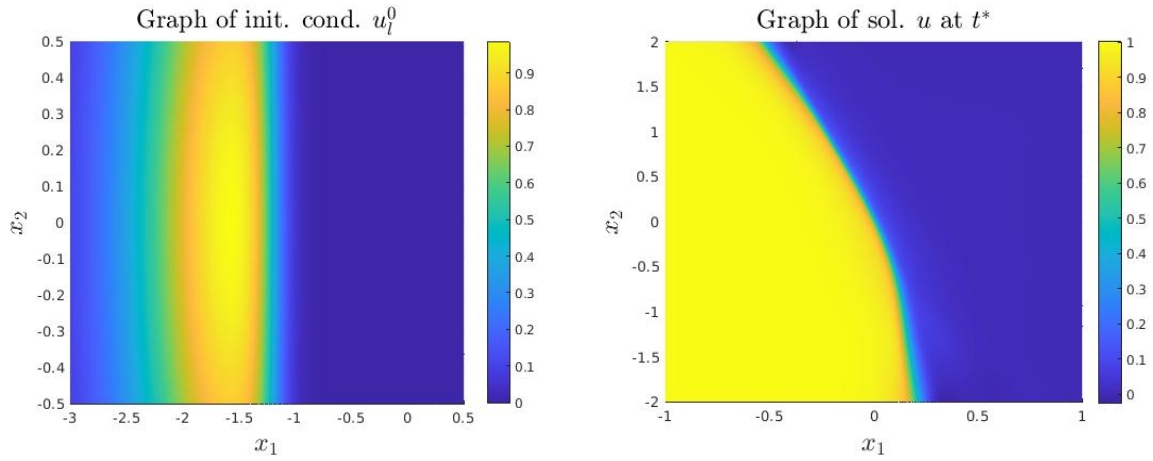

Figure 8: **Experiment 4.** (Left) Graph of u_0. (Right) Graph of the solution at $t = t^*$.

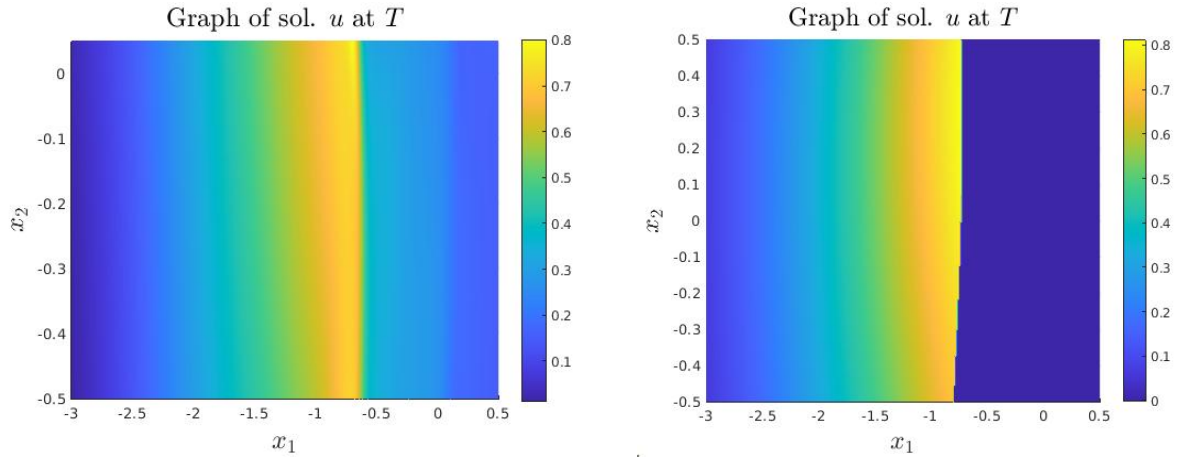

Figure 9: **Experiment 4.** (Left) Graph of the PINN solution at $t = T$. (Right) Graph of the leafnet solution at $t = T$.

variable reads $u = (u_1, u_2, u_3) = (\hbar, \hbar u, \hbar v)$, and the flux functions are defined as $F_k = (F_k^i)$, $k = 1, 2$, $i = 1, 2, 3$, where

$$\begin{aligned} F_k^*(u) &= (\hbar u, \quad \hbar u^2 + g\hbar^2/2, \quad \hbar uv) &&= (u_2, \quad u_2^2/u_1 + gu_1^2/2, \quad u_2u_3/u_1), \\ F_2^*(u) &= (\hbar v, \quad \hbar uv, \quad \hbar v^2 + g\hbar^2/2) &&= (u_3, \quad u_2u_3/u_1, \quad u_3^2/u_1 + gu_1^2/2). \end{aligned}$$

The system is strictly hyperbolic for $\hbar > 0$, and the eigenvalues of ∇F_1^* (written in nonconservative variables) are

$$\lambda_1^1 = u - \sqrt{g\hbar}, \quad \lambda_1^2 = u + \sqrt{g\hbar}, \quad \lambda_1^3 = u,$$

with corresponding eigenvectors

$$v_1^1 = (1, u - \sqrt{g\hbar}, v), \quad v_1^2 = (0, 0, 1), \quad v_1^3 = (1, u + \sqrt{g\hbar}, v).$$

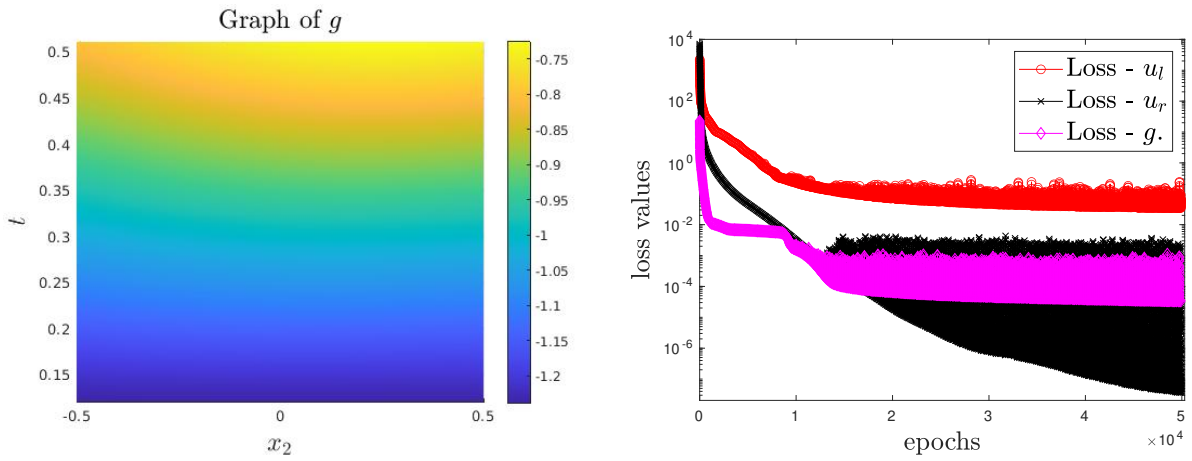

Figure 10: **Experiment 4.** (Left) Graph of the surface of discontinuity. (Right) Loss functions.

The eigenvalues of ∇F_2^* read (written in nonconservative variables)

$$\lambda_2^1 = v - \sqrt{g\hbar}, \quad \lambda_2^2 = v, \quad \lambda_2^3 = v + \sqrt{g\hbar},$$

and corresponding eigenvectors are given by

$$v_2^1 = (1, u, v - \sqrt{g\hbar}), \quad v_2^2 = (0, 1, 0), \quad v_2^3 = (1, u, v + \sqrt{g\hbar}).$$

Setting $\omega = (\omega_1, \omega_2) \in \mathbb{R}^2$ such that $|\omega| = 1$, the real eigenvalues of $\omega_1 \nabla F_1^* + \omega_2 \nabla F_2^*$ are given by $\ell_1(\omega) = \omega_1 u + \omega_2 v - \sqrt{g\hbar}$, $\ell_2(u, \omega) = \omega_1 u + \omega_2 v$, $\ell_3(\omega) = \omega_1 u + \omega_2 v + \sqrt{g\hbar}$. It will be convenient to solve the IBVP associated to (u_1, u_2, u_3) in nonconservative form, which in terms of $\hbar, u, v$ reads as

$$\begin{aligned} \partial_t \hbar + \hbar(\partial_{x_1} u + \partial_{x_2} v) + u\partial_{x_1}\hbar + v\partial_{x_2}\hbar &= 0\,, \\ \partial_t u + u\partial_{x_1} u + v\partial_{x_2} u + g\partial_{x_1}\hbar &= 0\,, \\ \partial_t v + u\partial_{x_1} v + v\partial_{x_2} v + g\partial_{x_2}\hbar &= 0\,. \end{aligned} \tag{82}$$

We assume here that the initial discontinuity is a straight line Γ_0 with equation $g_0(x_2) = \beta x_2$, where $\beta = 1/2$. We design a relevant non-trivial initial condition, we ensure that the left and right states at the discontinuity satisfy the Rankine-Hugoniot jump condition. In this goal, we first consider two *constant* conservative states $\overline{s}_0^l = (\overline{\hbar}_0^l, \overline{\hbar}_0^l \overline{u}_0^l, \overline{\hbar}_0^l \overline{v}_0^l)$ and $\overline{s}_0^r = (\overline{\hbar}_0^r, \overline{\hbar}_0^r \overline{u}_0^l, \overline{\hbar}_0^r \overline{v}_0^r)$, separated by the oblique shock Γ_0 with unit normal vector $\omega = (\omega_1, \omega_2)$, such that for some σ^0, we have

$$\sigma^0 [\![\overline{s}^0]\!] = \omega_1 [\![F_1(\overline{s}^0)]\!] + \omega_2 [\![F_2(\overline{s}^0)]\!]. \tag{83}$$

We assume here that the initial discontinuity is a straight line with $g_0(x_2) = \beta x_2$, where $\beta = -\omega_1/\omega_2$, so that from (83) we get

$$\frac{\sigma^0}{\omega_1} [\![\overline{s}^0]\!] = [\![F_1(\overline{s}^0)]\!] - \beta [\![F_2(\overline{s}^0)]\!].$$

Explicitly, we take $\omega_1 = 1/\sqrt{5}$, $\omega_2 = -2/\sqrt{5}$ corresponding to $\beta = 1/2$. After simple preliminary calculations, we select: $\overline{\hbar}_0^l = 2$, $\overline{\hbar}_0^r = 1$, $\overline{u}_0^l = \sqrt{2g/5}$, $\overline{u}_0^r = 0$, $\overline{v}_0^l = -\sqrt{g/10}$, $\overline{v}_0^r = 0$ and $\sigma^0 = \sqrt{5}\overline{u}_0^l/2 = \sqrt{g/2}$. The corresponding eigenvalues of $\omega_1 \nabla F_1(\overline{s}_\nu^0) + \omega_2 \nabla F_2(\overline{s}_\nu^0)$ are given by

$$\begin{array}{ll} \ell_1(\overline{s}_0^l) = \omega_1\big(\sqrt{2g/5} - 1\big) - \omega_2\big(\sqrt{g/10} + 1\big), & \ell_1(\overline{s}_0^r) = -\omega_1 - \omega_2, \\ \ell_2(\overline{s}_0^l) = \omega_1\sqrt{2g/5} - \omega_2\sqrt{g/10}, & \ell_2(\overline{s}_0^r) = 0, \\ \ell_3(\overline{s}_0^l) = \omega_1\big(\sqrt{2g/5} + 1\big) + \omega_2\big(1 - \sqrt{g/10}\big), & \ell_3(\overline{s}_0^r) = \omega_1 + \omega_2\,. \end{array}$$

Characterization of the shock. Selecting $g = 1$, and $\omega = (2, -1)/\sqrt{5}$, we then have $\sigma^0 = 1/\sqrt{2} \approx 0.7071$ and $\ell_1(\overline{s}_0^l) \approx 1.013$, $\ell_1(\overline{s}_0^r) \approx 0.447$, that is $\ell_1(\overline{s}_0^r) < \sigma^0 < \ell_1(\overline{s}_0^l)$ corresponding to a 1-shock.

Based on those preliminary computations we construct the initial data in nonconservative form $(\hbar_0^r, u_0^r, v_0^r) = (1, 0, 0)$ and $(\hbar_0^l, u_0^l, v_0^l)$ in Ω defined by

$$\begin{array}{rcl} \hbar_0^l(x_1, x_2) & = & (\overline{\hbar}_0^l - \zeta k_1(\min\{x_1, g_0(x_2)\}, x_2))k_2(\min\{x_1, g_0(x_2)\}, x_2), \\ u_0^l(x_1, x_2) & = & (\overline{u}_0^l - \zeta k_1(\min\{x_1, g_0(x_2)\}, x_2))k_2(\min\{x_1, g_0(x_2)\}, x_2), \\ v_0^l(x_1, x_2) & = & (\overline{v}_0^l k_1(\min\{x_1, g_0(x_2)\}, x_2))k_2(\min\{x_1, g_0(x_2)\}, x_2), \end{array}$$

and where

$$k_1(x_1, x_2) = g_0(x_2) - (x_1 - (a_1 + b_1)/2), \;\; k_2(x_1, x_2) = \gamma \tanh(\alpha k_1(x_1, x_2))\,.$$

Experiment 5 (2-D system, shock) Here we take $a_1 = -1$, $b_1 = 2$, $a_2 = -b_2 = -1$, $\alpha = 20$, $\zeta = 1/4$, $\gamma = 1/2$. The final T is fixed to 0.5. In the numerical experiment, the networks for approximating the IBVP on (u_1, u_2, u_3) and g have 2 hidden layers with 10 neurons. The number of randomly chosen collocation points is 150 in x_1, x_2 and t. We report in Fig. 11 the graphs of $\hbar^0$, u^0 and v^0. We then report in Fig. 12, the graphs of the reconstructed solution at final time T (nonconservative variables) $\hbar(\cdot, T)$, $u(\cdot, T)$ and $v(\cdot, T)$. We report in Fig. 13 (Left) the graph of g in Q_2 defining the time-space surface of discontinuity. For completeness, we also report the graph of convergence of the minimization algorithms in Fig. 13 (Right).

We yet observe that the propagation of the shock wave is performed without any artificial diffusion, which is expected from the design of the algorithm.

Experiment 5 bis (2-D system, shock) In this following experiment, we consider the exact same setting as above except that we now consider a non-straight initial discontinuity: $g_0(x_2) = \beta(x_2 - a_2)$, where $\beta = 10^{-1}$. Here, we report the initial and final leafnet solution in Fig. 14. We also report the correponding loss function and graph of $\mathbf{g}$ in Fig. 15

This test illustrates that the method can as well as non-oblique shock.

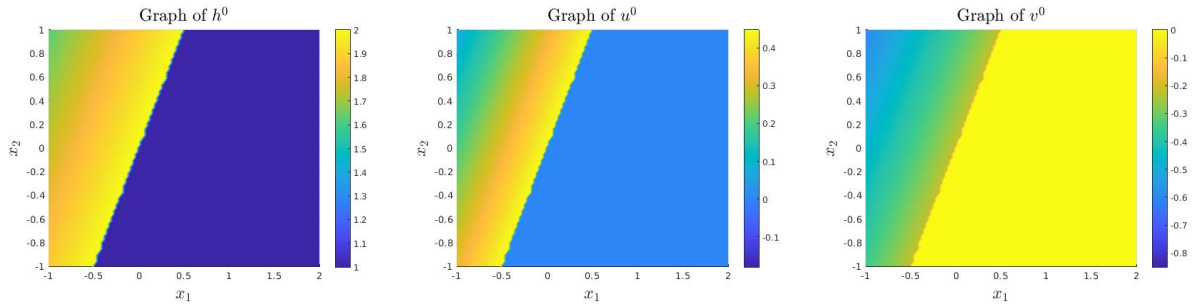

Figure 11: **Experiment 5.** (Left) Graph of $\hbar_l$. (Middle) Graph of u^0. (Right) Graph of v^0.

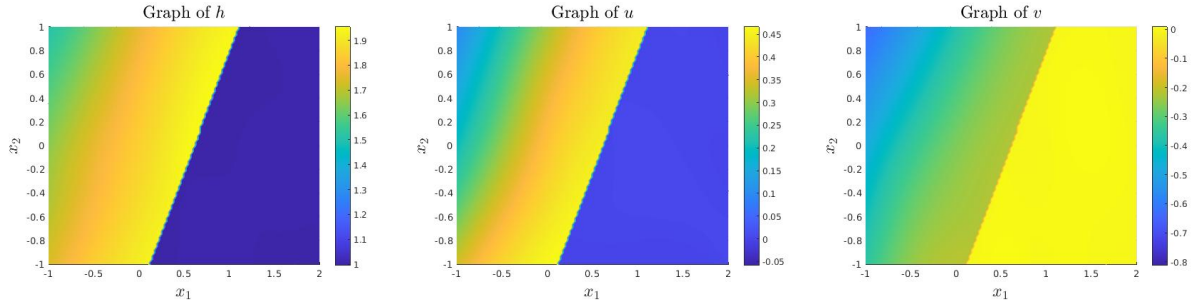

Figure 12: **Experiment 5.** At final time T (Left) Graph of $\boldsymbol{\hbar}(\cdot,\cdot,T)$. (Middle) Graph of $\boldsymbol{u}(\cdot,\cdot,T)$. (Right) Graph of $\boldsymbol{v}(\cdot,\cdot,T)$.

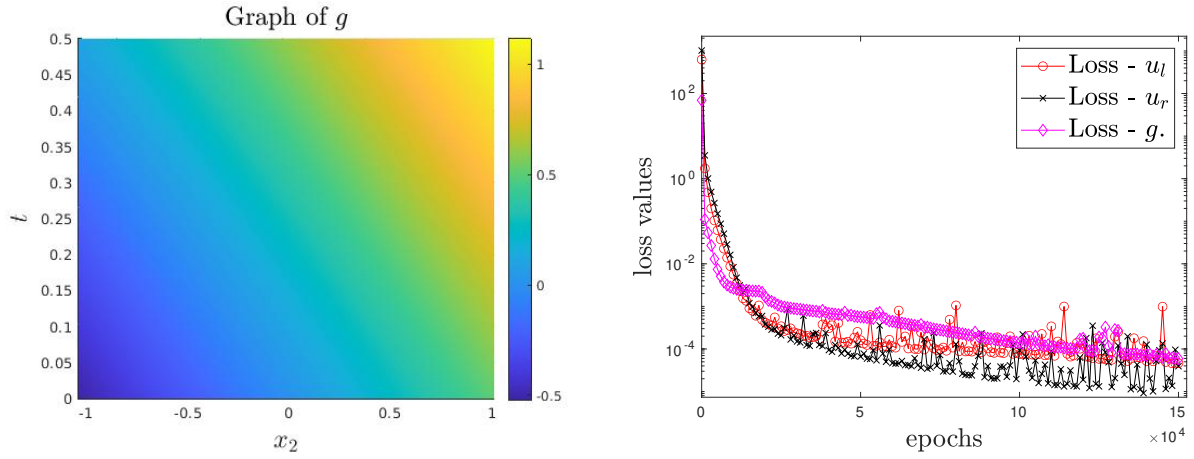

Figure 13: **Experiment 5.** (Left) Graph of $\mathbf{g}$. (Right) Loss functions for training $\mathbf{u}^l$, $\mathbf{u}^r$ and $\mathbf{g}$.

5. Conclusion

We have presented a neural network based computational method for accurately solving piece-wise smooth solutions to N-dimensional hyperbolic conservation laws, and more particularly for tracking shock waves, formation of shock waves and rarefaction waves. In the

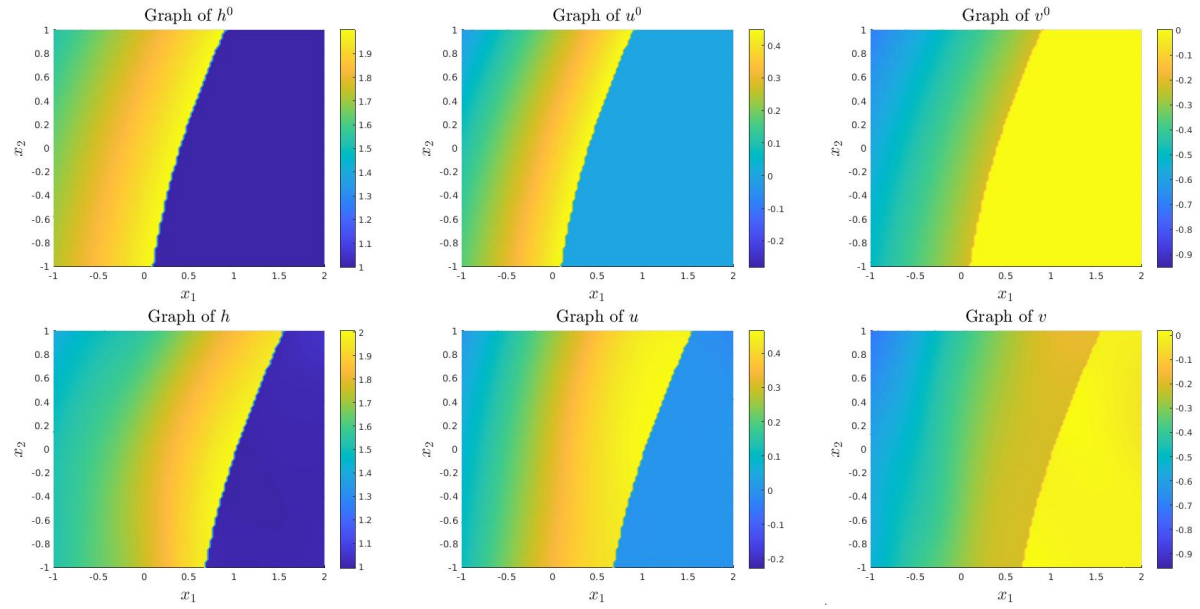

Figure 14: **Experiment 5 bis.** Initial data (Top) Leafnet solution at final time T (Bottom)At final time T.

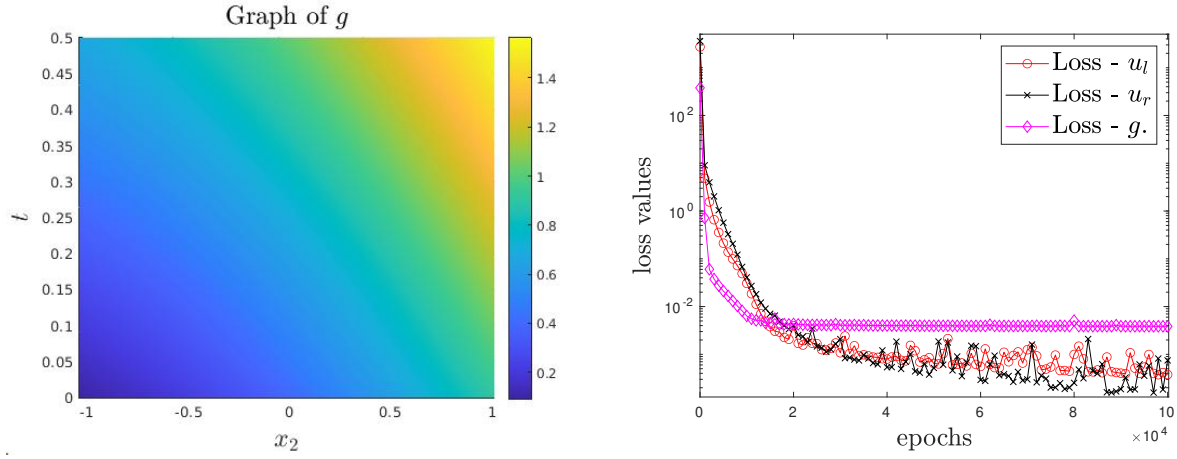

Figure 15: **Experiment 5 bis.** (Left) Graph of $\mathbf{g}$. (Right) Loss functions for training $\mathbf{u}^l$, $\mathbf{u}^r$ and $\mathbf{g}$.

case of shock waves, we have shown rigorously that the associated method is well-posed and we have given an error estimated of the shock wave approximation. We have shown that the proposed method can be extended to the formation and evolution of shock waves solutionto N-dimensional systems of conservation laws as well.

The driving idea of the method is the reformulation of the HCL as a coupled system involving only smooth local solutions, hence providing a mathematical setting compatible with the use of neural network algorithms approximating PDE in the their strong form. We then have developed a neural network-based algorithm for approximating the new equivalent

coupled system. Several numerical experiments were also proposed to illustrate the proposed strategy and its accuracy to approximate shock and rarefaction waves.

References

[1] D. Serre. *Systèmes de lois de conservation. I.* Fondations. [Foundations]. Diderot Editeur, Paris, 1996. Hyperbolicité, entropies, ondes de choc. [Hyperbolicity, entropies, shock waves].

[2] P. G. LeFloch. *Hyperbolic systems of conservation laws.* Lectures in Mathematics ETH Zürich. Birkhäuser Verlag, Basel, 2002. The theory of classical and nonclassical shock waves.

[3] J. Smoller. *Shock waves and reaction-diffusion equations*, volume 258 of *Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Science]*. Springer-Verlag, New York-Berlin, 1983.

[4] M. Raissi, P. Perdikaris, and G.E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.*, 378:686–707, 2019.

[5] G. Pang, L. Lu, and G.E. Karniadakis. fPINNs: fractional physics-informed neural networks. *SIAM J. Sci. Comput.*, 41(4):A2603–A2626, 2019.

[6] L. Yang, D. Zhang, and G. E. Karniadakis. Physics-informed generative adversarial networks for stochastic differential equations. *SIAM J. Sci. Comput.*, 42(1):A292–A317, 2020.

[7] I.E. Lagaris, A. Likas, and D.I. Fotiadis. Artificial neural networks for solving ordinary and partial differential equations. *IEEE Transactions on Neural Networks*, 9(5):987–1000, 1998.

[8] J. Han, A. Jentzen, and Weinan E. Solving high-dimensional partial differential equations using deep learning. *Proc. Natl. Acad. Sci. USA*, 115(34):8505–8510, 2018.

[9] H. Gao, M. J. Zahr, and J.-X. Wang. Physics-informed graph neural Galerkin networks: a unified framework for solving PDE-governed forward and inverse problems. *Comput. Methods Appl. Mech. Engrg.*, 390:Paper No. 114502, 18, 2022.

[10] J. Sirignano and K. Spiliopoulos. DGM: a deep learning algorithm for solving partial differential equations. *J. Comput. Phys.*, 375:1339–1364, 2018.

[11] F. Alouges, F. De Vuyst, G. Le Coq, and E. Lorin. The reservoir technique: a way to make godunov-type schemes zero or very low diffusive. application to Colella-Glaz. *Eur. J. Mech. B Fluids*, 27(6), 2008.

[12] P. L. Roe. Approximate Riemann solvers, parameter vectors, and difference schemes. *J. Comput. Phys.*, 43(2):357–372, 1981.

[13] E. Godlewski and P.-A. Raviart. *Numerical approximation of hyperbolic systems of conservation laws*, volume 118 of *Applied Mathematical Sciences*. Springer-Verlag, New York, 1996.

[14] E. Godlewski and P.-A. Raviart. *Hyperbolic systems of conservation laws*, volume 3/4 of *Mathématiques & Applications (Paris) [Mathematics and Applications]*. Ellipses, Paris, 1991.

[15] J.L. Montagne, H.C. Yee, and M. Yinokur. *Comparative study of high-resolution shock-capturing schemes for a real gas.*, volume 20) (ISBN 3-528-08094-9). 1987.

[16] E. Weinan and Bing Yu. The deep ritz method: A deep learning-based numerical algorithm for solving variational problems. *Communications in Mathematics and Statistics*, 6(1):1 – 12, 2018.

[17] E. Kharazmi, Z. Zhang, and G. E. Karniadakis. Variational physics-informed neural networks for solving partial differential equations, 2019.

[18] E. Lorin and A. Novruzi. Non-diffusive neural network method for hyperbolic conservation laws. *J. Comput. Phys.*, 513:113161, 2024.

[19] J.-M. Ghidaglia and F. Pascal. The normal flux method at the boundary for multidimensional finite volume approximations in CFD. *Eur. J. Mech. B Fluids*, 24(1):1–17, 2005.

[20] A. Majda. *Compressible Fluid Flow and Systems of Conservation Laws in Several Space Variables*. Springer-Verlag, 1984.

[21] T. De Ryck, S. Lanthaler, and S. Mishra. On the approximation of functions by tanh neural networks. *Neural Networks*, 143:732–750, 2021.

[22] T. De Ryck, A. D. Jagtap, and S. Mishra. Error estimates for physics-informed neural networks approximating the navier–stokes equations. *IMA Journal of Numerical Analysis*, 44(1):83–119, 2024.

[23] S. G. Krantz and H. R. Parks. *The Implicit Function Theorem: History, Theory, and Applications*. Modern Birkhäuser Classics. Birkhäuser / Springer Science+Business Media, New York, NY, 2013.